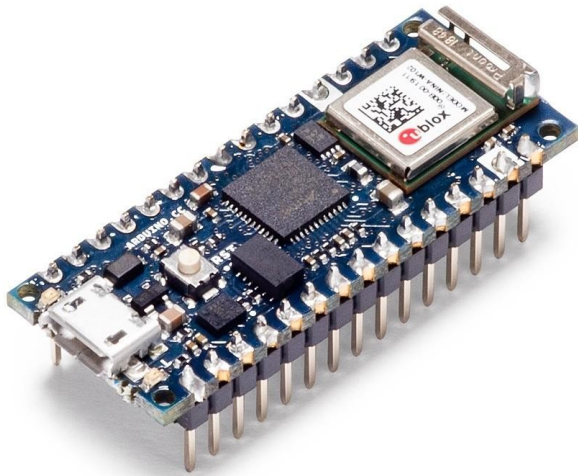




Arduino Nano 33 IoT, mit Headern



Artikel-Nr.:	ABX00032
Hersteller:	Arduino
EAN:	7630049201545
Herkunftsland:	Italien
Zolltarifnummer:	85423990
Gewicht:	0.018 kg

Der Arduino Nano 33 IoT ist der einfachste und günstigste Einstiegspunkt, um bestehende Geräte zu verbessern (und neue zu erstellen), um Teil des IoT zu werden und Pico-Netzwerk-Anwendungen zu entwerfen. Egal, ob Sie ein Sensornetzwerk aufbauen wollen, das mit Ihrem Büro- oder Heim-Router verbunden ist, oder ob Sie ein BLE-Gerät erstellen wollen, das Daten an ein Mobiltelefon sendet, der Nano 33 IoT ist Ihre Komplettlösung für viele der grundlegenden IoT-Anwendungsszenarien.

Der Hauptprozessor des Boards ist ein stromsparender Arm® Cortex®-M0 32-bit SAMD21. Die WiFi- und Bluetooth®-Konnektivität wird mit einem Modul von u-blox, dem NINA-W10, realisiert, einem Low-Power-Chipsatz, der im 2,4-GHz-Bereich arbeitet. Darüber hinaus wird die sichere Kommunikation durch den Microchip® ECC608 Krypto-Chip gewährleistet. Außerdem finden Sie eine 6-Achsen-IMU, was dieses Board perfekt für einfache Vibrationsalarmsysteme, Schrittzähler, relative Positionierung von Robotern, etc. macht.

Arduino IoT Cloud

Verwenden Sie Ihr Nano-Board auf der IoT-Cloud von Arduino, einer einfachen und schnellen Möglichkeit, eine sichere Kommunikation für alle Ihre angeschlossenen Dinge zu gewährleisten.

Offizielle Arduino WiFi Library

Bei Arduino haben wir die Verbindung zu einem WiFi-Netzwerk so einfach gemacht wie das Blinken einer LED. Sie können Ihr Board dazu bringen, sich mit jeder Art von existierendem WiFi-Netzwerk zu verbinden, oder es benutzen, um Ihren eigenen Arduino Access Point zu erstellen. Die spezifischen Beispiele, die wir für den Nano 33 IoT bereitstellen, können Sie auf der [Referenzseite zur WiFININA-Bibliothek](#) nachlesen.

Kompatibel mit anderen Cloud-Diensten

Es ist auch möglich, Ihr Board mit verschiedenen Cloud-Diensten zu verbinden, unter anderem mit dem Arduino-eigenen. Hier einige Beispiele, wie man die Arduino-Boards dazu bringt, sich zu verbinden:

- **Blynk:** ein [einfaches Projekt](#) aus unserer Community, das sich mit Blynk verbindet, um Ihr Board mit wenig Code von einem Telefon aus zu bedienen
- **IFTTT:** ein ausführliches Beispiel für den [Bau eines intelligenten Steckers](#), der mit IFTTT verbunden ist
- **AWS IoT Core:** wir haben [dieses Beispiel](#) erstellt, wie man sich mit Amazon Web Services verbindet
- **Azure:** besuchen Sie [dieses Github-Repository](#), das erklärt, wie man einen Temperatursensor mit der Azure-Cloud verbindet
- **Firebase:** wenn Sie sich mit Googles Firebase verbinden möchten, zeigt Ihnen [diese Arduino-Bibliothek](#), wie es geht

Hinweis: Während die meisten der oben gezeigten Beispiele auf dem MKR WiFi 1010 laufen, haben beide Boards den gleichen Prozessor und Wireless-Chipsatz, was bedeutet, dass es möglich sein wird, sie mit dem Nano 33 IoT zu replizieren.

Bluetooth® und BLE

Der Kommunikations-Chipsatz auf dem Nano 33 IoT kann sowohl ein BLE- und Bluetooth®-Client als auch ein Host-Gerät sein. Etwas



ziemlich Einzigartiges in der Welt der Mikrocontroller-Plattformen. Wenn Sie sehen wollen, wie einfach es ist, eine Bluetooth®-Zentrale oder ein Peripheriegerät zu erstellen, erkunden Sie die Beispiele in unserer [ArduinoBLE-Bibliothek](#).

Offen, damit Sie mithacken können

Der Nano 33 IoT ist ein Dual-Prozessor-Gerät, das zum Experimentieren einlädt. Durch das Hacken des WiFinINA-Moduls können Sie z.B. sowohl WiFi als auch BLE / Bluetooth® gleichzeitig auf dem Board nutzen. Eine weitere Möglichkeit ist, eine superleichte Version von Linux auf dem Modul laufen zu lassen, während der Haupt-Mikrocontroller Low-Level-Geräte wie Motoren oder Bildschirme steuert. Diese experimentellen Techniken erfordern fortgeschrittenes Hacken auf Ihrer Seite. Sie sind möglich, indem Sie die Firmware des Moduls modifizieren, die Sie in [unseren Github-Repositories](#) finden können.

VORSICHT: diese Art von Hacking bricht die Zertifizierung Ihres WiFinINA-Moduls, tun Sie es auf eigene Gefahr.

Weitere Bilder:

