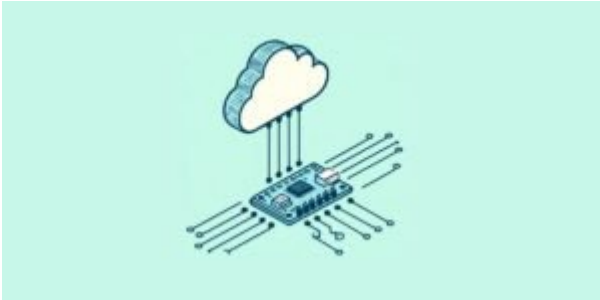


Die Arduino Cloud – erste Schritte



Bei Pollux Labs findest du zahlreiche Projekte, die sich mit IoT auseinandersetzen – seien es ein [ESP8266 Webserver](#), [Nachrichten vom Microcontroller](#) oder die [smartphonegesteuerte Pflanzbewässerung](#).

Hierbei kommen immer spezifische Entwicklungen zum Einsatz, mit deren Hilfe du dein jeweiliges Ziel umsetzt. Aber es geht natürlich auch komfortabler: Die Arduino Cloud ist ein Ökosystem, mit dem du deine Microcontroller vernetzen, auf Dashboards aktuelle Messdaten verfolgen und dich bei von dir festgelegten Ereignissen auf deinem Smartphone informieren lassen kannst. Das funktioniert nicht nur mit Arduinos – auch deinen ESP8266 oder ESP32 kannst du damit bequem ins Internet of Things bringen.

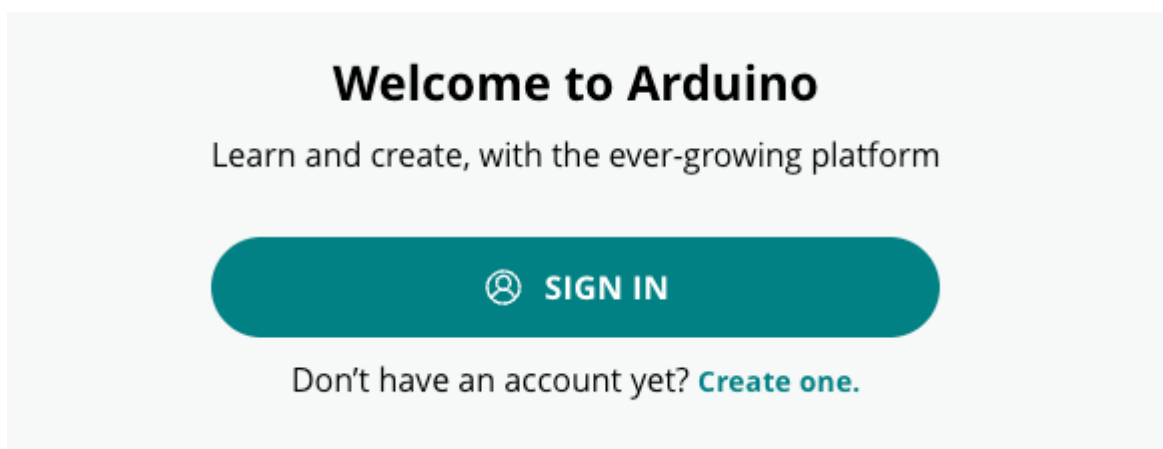
Außerdem bist du dabei **nicht auf dein eigenes WLAN beschränkt** und musst dich mit einer möglicherweise riskanten Portweiterleitung auseinandersetzen: **Du kannst von überall auf deine verbundenen Geräte zugreifen und sie steuern.**

In diesem Tutorial erfährst du, wie du

- einen kostenlosen Account in der Arduino Cloud erstellst,
- deinen ESP8266 dort registrierst, ihn programmierst und
- ihn von deinem Smartphone aus steuerst.

Einen Account in der Arduino Cloud erstellen

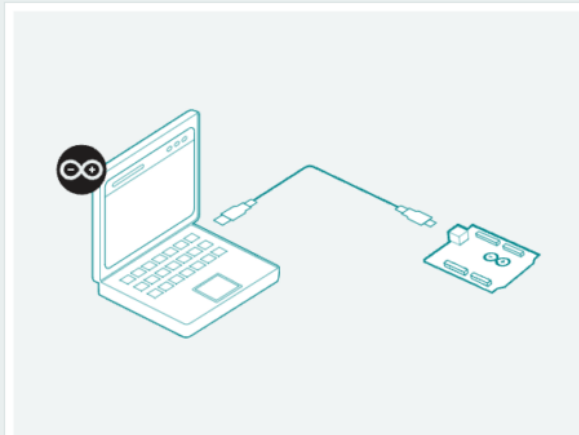
Bevor es losgehen kann, benötigst du ein **kostenloses Benutzerkonto**. Besuche hierfür die Webseite der [Arduino Cloud](#) und klicke oben rechts auf **Get started**. Wähle dann den Link **Create one**. Falls du bereits einen Account hast, kannst du dich natürlich umgehend einloggen.



Den Arduino Create Agent installieren

Nachdem du deinen Account erstellt hast, benötigst du noch ein kleines Plugin. Der **Arduino Create Agent** sorgt für die Kommunikation deines Microcontrollers am PC mit der Arduino Cloud. Möglicherweise wurdest du schon nach deinem ersten Login mit einem Popup darauf aufmerksam gemacht – falls nicht, du kannst ihn [auf dieser Webseite herunterladen](#) und ihn anschließend installieren.

WELCOME TO THE ARDUINO CREATE AGENT INSTALLATION!



You're about to begin the process of downloading and installing the Arduino Create Agent. The agent will provide you with several features:

- Recognize Arduino boards and other supported devices connected to your computer via USB;
- Upload sketches from your web browser to your boards via USB or through a network;
- Read data from serial monitor, as well as write to it.

START

Sobald die Installation abgeschlossen ist, starte das Plugin – es muss übrigens immer laufen, wenn du Microcontroller mit der Arduino Cloud programmierst, also einen Sketch hochladen möchtest.

Dein erstes Board hinzufügen

Wenn du diese beiden grundlegenden Schritte geschafft hast, wird es Zeit, deinen ersten Microcontroller zur Cloud hinzuzufügen. Im Folgenden dient der ESP8266 als Beispiel.

Klicke dafür im Menü links auf **Devices** und anschließend oben rechts auf **+ Device**. Auf der folgenden Seite kannst du auswählen, welches Gerät du verbinden möchtest – ein Arduino Board, einen Microcontroller eines „Drittanbieters“ oder etwas, um dessen Integration du dich selbst kümmerst (Manual). Der ESP8266 fällt in die zweite Kategorie, wähle als in diesem Fenster **Third party device**.

Im nächsten Schritt kannst du nun also den ESP8266 anhängen und das Modell auswählen:

← Setup Device X

Select device type

Please select the device type and model you want to configure

☒ ESP8266 ☐ ESP32 ☐ LoRaWAN

Select model ▼

- Phoenix 2.0
- Phoenix 1.0
- Digistump Oak
- NodeMCU 1.0 (ESP-12E Module)**
- NodeMCU 0.9 (ESP-12 Module)
- Olimex MOD-WIFI-ESP8266(-DEV)
- Adafruit Feather HUZZAH ESP8266

Fehlt nur noch ein Name für das Board, den du gleich danach vergeben kannst. Hier empfiehlt es sich, auf den Verwendungszweck deines Controllers einzugehen, damit du ihn in einer längeren Liste leicht wiederfinden kannst.

Nun noch ein wichtiger Schritt: Du erhältst eine **Device ID** und einen **Secret Key** – zwei Zeichenfolgen, die du dir sicher abspeichern musst, da du sie später wieder benötigst. Praktischerweise kannst du dir sie auch als übersichtliches PDF herunterladen.

Sobald du eine Erfolgsmeldung erhältst, ist dein ESP8266 in

der Arduino Cloud registriert und taucht in der entsprechenden Device-Liste auf.

Ein Thing erstellen

Klicke nun als nächstes auf den Listeneintrag des ESP8266. Es öffnet sich ein Fenster, in dem du auf der linken Seite einige Daten siehst und darunter den Button **Create thing**.



NodeMCU 1.0 (ESP-12E Module)
● Offline

Last Activity -

Added 15. Mai 2024, 12:59:34

ID bb9ef003-0b60-48b7-b49c-86e42f99f8dd

FQBN esp8266:esp8266:nodemcuv2

Associated Thing

No Thing associated with ESP8266-1

Create thing

Ein sogenanntes **Thing** ist so etwas wie das Abbild des ESP8266 in der Arduino Cloud. Hier kannst du seine Verbindung zu deinem WLAN einrichten, indem du die Zugangsdaten hinterlegst, die dann auf deinem ESP8266 abgespeichert werden, sobald du einen Sketch auf ihn hochlädst.

Außerdem kannst du Variablen erstellen, die sowohl im Sketch Verwendung finden als auch später in ein Dashboard eingebunden werden können. Ein Thing ist also so etwas wie eine Ebene zwischen Cloud und Hardware. Klicke nun als nächstes rechts im Bereich **Netword** auf **Configure**.

Cloud Variables

ADD

Variables are what you can monitor or control to make your Thing function. For example a temperature or a smart lamp. Once created, you can use them in your Sketch.



Associated Device

ESP8266-1

ID: bb9ef003-0b60-48b7-b49c-...

Type: NodeMCU 1.0 (ESP-12E Module)

Status: Offline



Change



Detach

Network

Enter your network credentials to connect your device.



Configure

Smart Home integration

Configure your Thing to work with Amazon Alexa or Google Home



Configure

Trage nun deine WLAN-Zugangsdaten sowie deinen **Secret Key** ein und speichere die Daten ab. Wenn du wieder zurück in der Übersicht bist, klicke oben im Bereich **Cloud Variables** auf **Add**. Hier hinterlegst du nun deine erste Variable.

Als ersten Test wirst du deinen ESP8266 so einrichten, dass du seine interne LED von einem Dashboard aus ein- und ausschalten kannst. Hierfür benötigst du eine entsprechende Variable, in der gespeichert ist, ob die LED leuchtet oder nicht. Hierfür reicht der Typ **bool**, der entweder den Wert **true** oder **false** annehmen kann. Stelle deine erste Variable also wie folgt ein:

Name
internalLED

 **Sync with other Things** ⓘ

Boolean eg. true ▼

Declaration
`bool internalLED;` ⓘ

Variable Permission ⓘ

☒ Read & Write ☐ Read Only

Variable Update Policy ⓘ

☒ On change ☐ Periodically

CANCEL

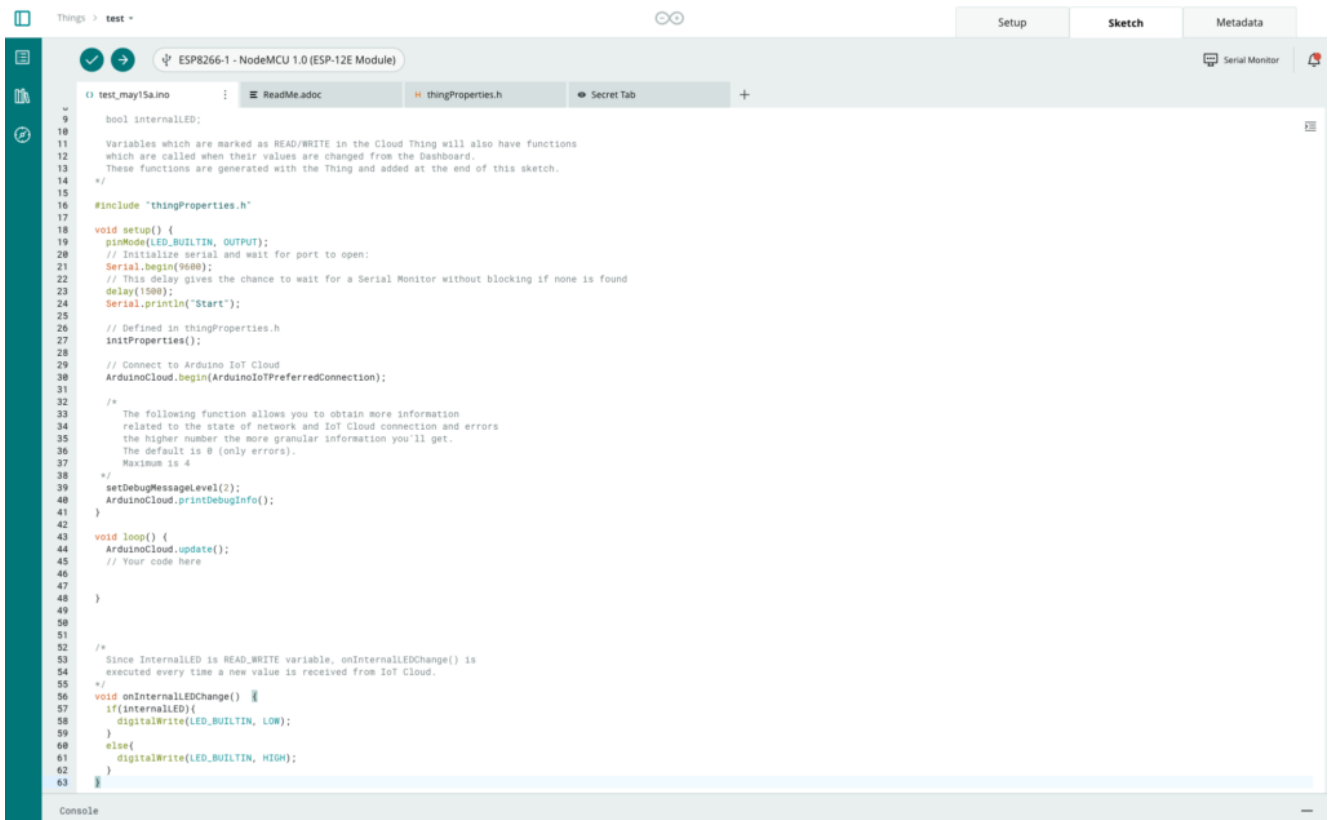
ADD VARIABLE

Klicke abschließend auf den Button **Add Variable**.

Der erste Sketch für die Steuerung der

LED

Jetzt folgt der Sketch, den du auf deinen ESP8266 hochladen wirst. Oben rechts findest du den Reiter **Sketch**. Nach einem Klick hierauf öffnet sich der Editor – ähnlich der Arduino IDE, die du sicherlich schon kennst:



Ein Großteil des Codes ist hier schon ausgefüllt, z.B. der Verweis auf die Zugangsdaten zu deinem WLAN oder der Start des Seriellen Monitors. Damit du die interne LED jedoch steuern kannst, fehlt noch ein bisschen. Zunächst der Befehl **pinMode()** für die Variable **LED_BUILTIN** (also die interne LED). Platziere diese Zeile Code in der Funktion **void setup()**.

```
pinMode(LED_BUILTIN, OUTPUT);
```

Außerdem musst du noch die Funktion **onInternalLEDChange()** anpassen:

```
void onInternalLEDChange() {
```



```
if(internalLED){  
    digitalWrite(LED_BUILTIN, LOW);  
}  
else{  
    digitalWrite(LED_BUILTIN, HIGH);  
}  
}
```

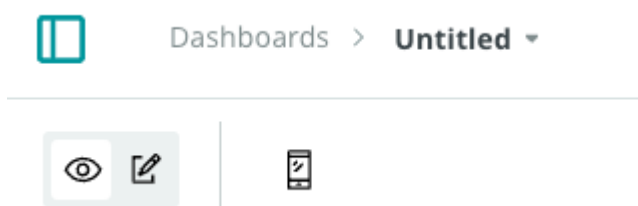
Hier legst du fest, dass wenn die Variable **internalLED** den Wert **true** hat, die interne LED ausgeschaltet wird. Das fühlt sich zwar „falsch herum“ an, ist aber das normale Verhalten des ESP8266... Wenn der Wert auf **false** springt, geht sie hingegen an.

Sobald du diese zwei Stellen ausgebessert hast, kannst du den Sketch auf deinen ESP8266 hochladen. Klicke hierfür links oben auf den Pfeil nach rechts. Hat alles geklappt? Dann wird es Zeit für den letzten Teil dieses Tutorials.

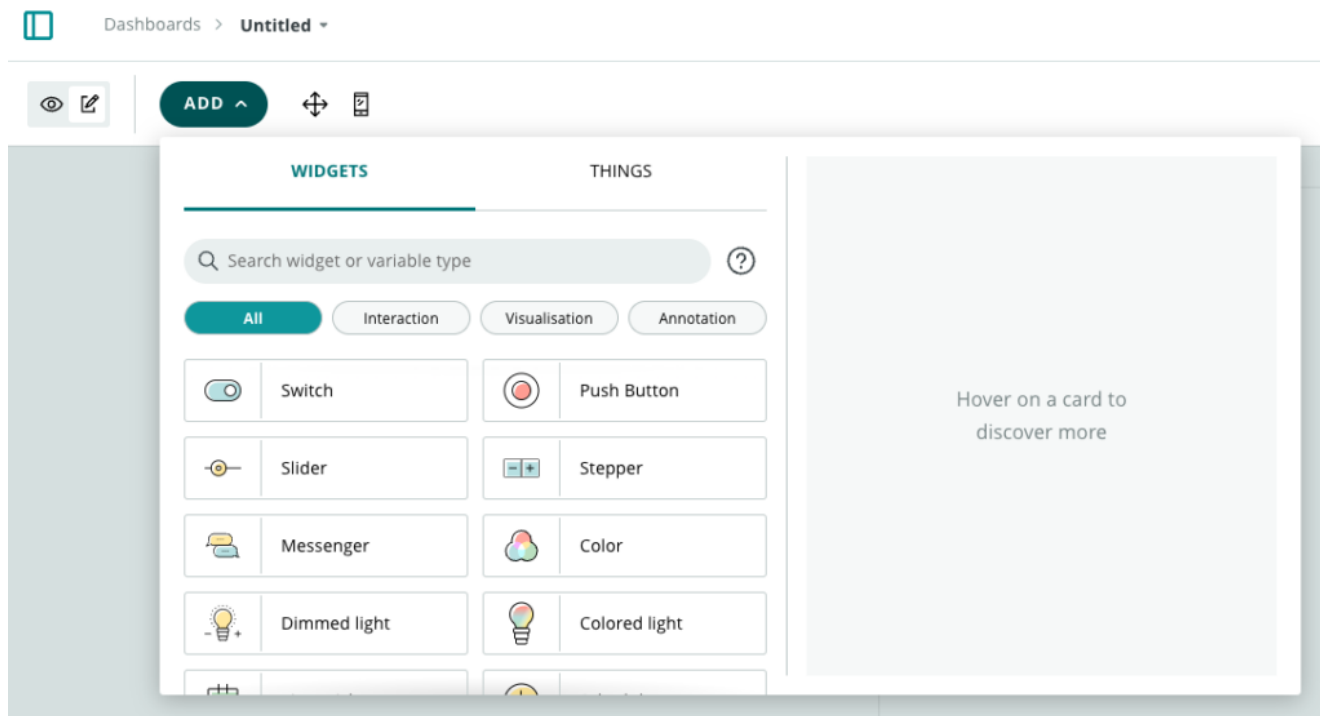
Die LED vom Dashboard (und vom Smartphone) aus steuern

Klicke im Menü links auf **Dashboards** und anschließend oben rechts auf **+ Dashboard**. Anschließend landest du auf einer leeren Seite – nur oben findest du ein paar Buttons. Die leere Fläche ist dein frisches Dashboard, auf dem du Elemente wie Schalter, Diagramme, Buttons etc. platzieren kannst.

An dieser Stelle integrierst du einen Schalter, der die LED des ESP8266 an- und ausschalten kann. Klicke hierfür zunächst oben links auf den Bleistift:



Nun findest du rechts daneben den Button **Add**. Klicke hierauf, um die Übersicht der verfügbaren Widgets zu öffnen:



Wähle hier nun die Option **Switch** – das wird der Schalter für die LED deines ESP8266. Anschließend landet der Schalter auf die Mitte des Dashboards. Fehlt nur noch die Verbindung zur LED, sprich zur Variablen **internalLED**, die du mit diesem Schalter zwischen **true** und **false** hin- und herspringen lassen kannst.

Klicke dafür im rechten Bereich des Fenster unter **Linked Variable** auf den grünen Button und wähle die Variable aus:



Fehlt nur noch ein Klick auf den Button **Link Variable** – und anschließend unten rechts auf **Done**. Anschließend befindest du dich wieder auf deinem Dashboard, dessen Editor-Modus du oben links mit einem Klick auf das Auge verlassen kannst.

Nun ist dein Schalter betriebsbereit. Vorausgesetzt, dein ESP8266 läuft (entweder mit Strom vom USB-Kabel an deinem PC oder von einer anderen Quelle) und ist mit dem WLAN verbunden, kannst du nun mit dem Schalter die interne LED an- und ausschalten.

Den Schalter von unterwegs bedienen

Dein Dashboard betrachtest du gerade im Browser auf deinem Computer – aber du kannst den ESP8266 von unterwegs mit deinem Smartphone steuern! Hierfür benötigst du die [IoT Remote App](#) der Arduino Cloud, die es für Android und iOS gibt.

Nachdem du dich in der App angemeldet hast, findest du auch dort dein gerade erstelltes Dashboard mit dem Schalter. Und wie zu erwarten, kannst du auch damit deinen ESP8266 steuern – **natürlich auch von unterwegs, ohne im heimischen WLAN zu sein.**

Wie geht es nun weiter? Mit den Grundlagen, die du in diesem Tutorial gelernt hast, steht dir der Weg offen für zahlreiche IoT-Projekte – sei es mit einem Microcontroller oder auch mit mehreren, die du über die Arduino Cloud miteinander vernetzt. In Zukunft wirst du hier auf Pollux Labs noch viele weitere Ideen und Projekte finden. Bis dahin, viel Erfolg!