



## Getting Started With the Arduino IoT Cloud

[Cloud IoT Arduino](#) > Getting Started With the Arduino IoT Cloud

Introduction

IoT Cloud Documentation

Features

Compatible Hardware

Wi-Fi

LoRaWAN®

GSM / NB-IoT Boards

ESP32 / ESP8266

Support

A Walk Through the Configuration

1. Creating an Arduino Account

2. Go to the Arduino IoT Cloud

3. Creating a Thing

4. Configuring a Device

5. Creating Variables

6. Connecting to a Network

7. Editing the Sketch

8. Creating a Dashboard

Congratulations

More tutorials

# Getting Started With the Arduino IoT Cloud

The Arduino IoT Cloud is a online platform that makes it easy for you to create, deploy and monitor IoT projects.

Ⓢ AUTHOR: **Karl Söderby** 📄 LAST REVISION: **14/09/2022, 10:09**

## Introduction

Connected devices around the world are increasing by billions every year. The [Arduino IoT Cloud](#) is a platform that allows **anyone** to create IoT projects, with a user friendly interface, and an all in one solution for **configuration, writing code, uploading and visualization.**

In this article, we will take a look at some different components of the Arduino IoT Cloud, and provide a general overview.

But if you're itching to get started and explore the Arduino IoT Cloud yourself, that is also perfectly fine! You can always come back here for more information!

[Go to Arduino IoT Cloud](#)

## IoT Cloud Documentation

Missing something?

Check out our store and get what you need to follow this tutorial.

[VISIT OUR STORE](#)

Suggest Changes

The content on docs.arduino.cc is facilitated through a public GitHub repository. You can read more on how to contribute in the [contribution policy](#).

[EDIT THIS PAGE](#)

Ayuda



To find **all tutorials & articles**, visit the [Arduino IoT Cloud Documentation page](#).

For a **technical overview, list of features**, and **API guide**, visit the [Arduino IoT Cloud Cheat Sheet](#).

For **API & SDK Documentation**, visit the developer reference at [Arduino IoT Cloud API](#).

## Features

Below is a list of Arduino IoT Cloud features.

**Data Monitoring** - learn how to easily monitor your Arduino's sensor values through a dashboard.

**Variable Synchronisation** - variable synchronisation allows you to sync variables across devices, enabling communication between devices with minimal coding.

**Scheduler** - schedule jobs to go on/off for a specific amount of time (seconds, minutes, hours).

**Over-The-Air (OTA) Uploads** - upload code to devices not connected to your computer.

**Webhooks** - integrate your project with another service, such as IFTTT.

**Amazon Alexa Support** - make your project voice controlled with the Amazon Alexa integration.

**Dashboard Sharing** - share your data with other people around the world.

## Compatible Hardware



ESP32 / ESP8266 microcontroller. The Arduino IoT Cloud currently supports connection via Wi-Fi, LoRaWAN® (via The Things Network) and mobile networks.

**i** All cloud-compatible Arduino boards come with a hardware secure element (such as the [ECC508](#) cryptochip), where you can store your security keys.

## Wi-Fi

**i** Official Arduino boards only supports the 2.4GHz frequency band for transmitting data.

The following boards connect to the Arduino IoT Cloud via Wi-Fi.

[MKR 1000 WiFi](#)

[MKR WiFi 1010](#)

[Nano RP2040 Connect](#)

[Nano 33 IoT](#)

[Portenta H7](#)

[Portenta H7 Lite Connected](#)

[Nicla Vision](#)

Connection via Wi-Fi is an easy alternative, and your credentials can safely be entered during the configuration of a project. This type of connection is most suitable for low-range projects, where you connect your board to the cloud via your home/work/school router.



a LoRaWAN® Network Server connected to thousands of public LoRa® gateways.

[MKR WAN 1300](#)

[MKR WAN 1310](#)

Connection via LoRaWAN® is recommended for low-power projects in both remote and urban areas, where Wi-Fi or other popular connectivity types are not available. The MKR WAN 1300/1310 boards are equipped with a LoRa radio module and has a slot for an antenna. With the right low-power configuration, the board can send data to the cloud for months on a single battery.



To learn more about setting up LoRaWAN® devices, visit the [Configuring LoRaWAN® devices in the Arduino Cloud](#) guide.

## GSM / NB-IoT Boards

The MKR GSM 1400 and MKR NB 1500 require a **SIM card** to connect to the cloud, as they communicate over the mobile networks.

[MKR GSM 1400](#)

[MKR NB 1500](#)

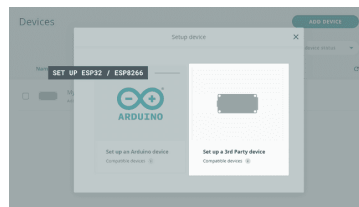
Connection through mobile networks can be considered in remote areas where there's no Wi-Fi, or in mobile projects (such as cargo tracking).



**i** Note that a secured connection is a memory intense operation, so there's not a lot of memory for the user application (e.g. around 2.6 kB on the MKR GSM 1400). Using a lot of IoT Cloud variables may cause the sketch to run out of memory on boards which don't offload the SSL stack and make it crash.

## ESP32 / ESP8266

The Arduino IoT Cloud supports a wide range of third party boards based on the ESP32 and ESP8266 microcontrollers with support for Wi-Fi. To set them up, simply choose the **third party option** in the device setup.



Configuring third party boards.

**i** To learn more about ESP32/ESP8266 support and how to set it up, visit the [Connecting ESP32 & ESP8266 to Arduino Cloud IoT guide](#).

## Support



features in the **Arduino Help Center**. If you don't find the answer you are looking for, we are always happy to help you with any question regarding our products!

[Go to Arduino Help Center](#)

## A Walk Through the Configuration



Setting up the Arduino IoT Cloud.

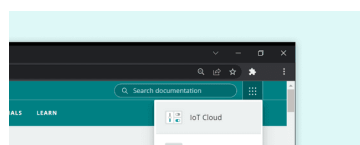
Setting up the Arduino IoT Cloud and accessing the different features available involves a few simple steps. So let's take a look at how to go from start to finish!

### 1. Creating an Arduino Account

To starting using the Arduino IoT cloud, we first need to [log in or sign up to Arduino](#).

### 2. Go to the Arduino IoT Cloud

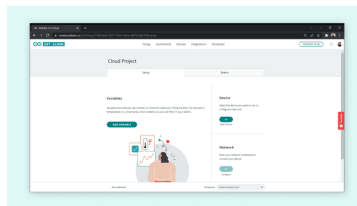
After we have signed up, you can access the Arduino IoT Cloud from any page on [arduino.cc](#) by clicking on the four dots menu in the top right corner. You can also [go directly to the Arduino IoT Cloud](#).





### 3. Creating a Thing

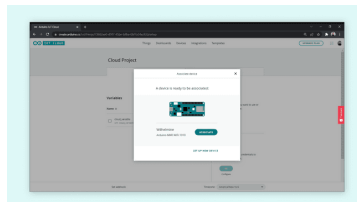
The journey always begin by creating a new **Thing**. In the Thing overview, we can choose what device to use, what Wi-Fi network we want to connect to, and create variables that we can monitor and control. This is the main configuration space, where all changes we make are automatically generated into a **special sketch file**.



The Thing overview.

### 4. Configuring a Device

Devices can easily be added and linked to a Thing. The Arduino IoT Cloud requires your computer to have the [Arduino Agent installed](#). The configuration process is quick and easy, and can be done by clicking on the **“Select device”** button in the Thing overview. Here, we can choose from any board that has been configured, or select the **“Configure new device”** option.



Configuring a device.

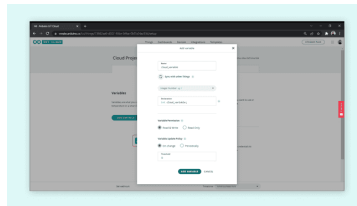
We can also get a complete overview of our devices by clicking the **“Devices”** tab at the top of the Arduino IoT Cloud interface. Here we can manage and add new devices.



The device tab.

## 5. Creating Variables

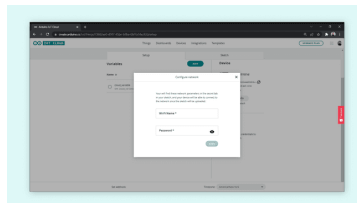
The variables we create are automatically generated into a sketch file. There are several data types we can choose from, such as **int**, **float**, **boolean**, **long**, **char**. There's also special variables, such as **Temperature**, **Velocity**, **Luminance** that can be used. When clicking on the **"Add variable"** button, we can choose name, data type, update setting and interaction mode.



Creating variables.

## 6. Connecting to a Network

To connect to a Wi-Fi network, simply click the **"Configure"** button in the network section. Enter the credentials and click **"Save"**. This information is also generated into your sketch file!



Entering network credentials.

## 7. Editing the Sketch

Now that we have configured variables, devices and network settings, we can get to programming our devices!



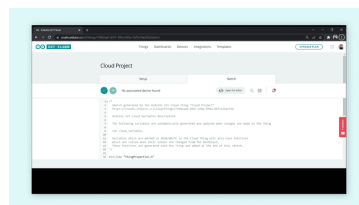


me, but with some additional code to make the connection to your network and to the cloud.

A sketch that, for example, reads an analog sensor, and use the **cloud variable** to store it. When the sketch has been uploaded, it will work as a regular sketch, but it will also update the cloud variables that we use!

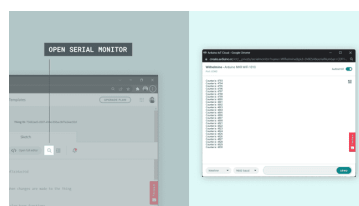
Additionally, each time we create a variable that has the **Read & Write** permission enabled, a function is also generated, at the bottom of your sketch file. Each time this variable changes, it will execute the code within this function! This means that we can leave most of the code out of the **loop()** and only run code when needed.

To upload the program to our board, simply click the **"Upload"** button.



Editing a sketch in the cloud editor.

The editor also has a **Serial Monitor Tool**, which can be opened by clicking the magnifying glass in the toolbar. Here you can view information regarding your connection, or commands printed via `Serial.print()` .



The Serial Monitor Tool.



successful, it will print **“connected to network\_name”** and **“connected to cloud”**. If it fails to connect, it will print the errors here as well.

**i** The cloud editor is a mirrored "minimal" version of the Web Editor. Any changes you make will also be reflected in the Web Editor, which is more suitable for developing more advanced sketches.

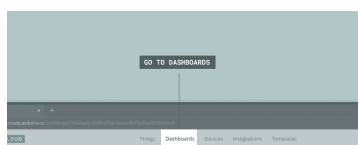
## 8. Creating a Dashboard

Now that we have configured the device & network, created variables, completed the sketch and successfully uploaded the code, we can move on to the fun part: **creating dashboards!**



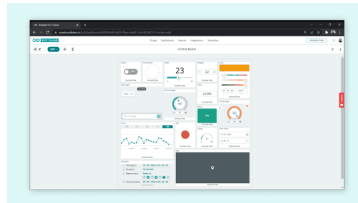
Visualize your data.

Dashboards are visual user interface for interacting with your boards over the cloud, and we can setup many different setups depending on what your IoT project needs. We can access our dashboards by clicking on the **“Dashboards”** tab at the top of the Arduino IoT Cloud interface, where we can create new dashboards, and see a list of dashboards created for other Things.





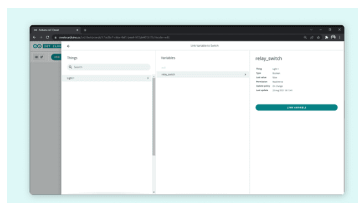
If we click on **"Create new dashboard"**, we enter a dashboard editor. Here, we can create something called **widgets**. Widgets are the visual representation of our variables we create, and there are many different to choose from. Below is an example using several types of widgets.



The different widgets available.

When we create widgets, we also need to **link them to our variables**. This is done by clicking on a widget we create, select a Thing, and select a variable that we want to link. Once it is linked, we can either interact with it, for example a button, or we can monitor a value from a sensor. As long as our board is connected to the cloud, the values will update!

Let's say we have a **temperature widget** that we want to link to the **temperature** variable inside the **Cloud project** thing.



Linking a variable to a widget.



example not be linked, and will not be an option while setting up your dashboard.

We can also have several things running at once, depending on your Arduino IoT Cloud plan, which we can include in the same dashboard. This is a great feature for tracking multiple boards in for example a larger sensor network, where boards can be connected to different networks around the world, but be monitored from the same dashboard.

## Congratulations



What will you build?

You are just a few easy steps from deploying your very own IoT project. Having a project connected to the Arduino IoT Cloud opens up many possibilities, such as tracking data in real time, triggering remote devices and building wireless systems.

What will you create?

[Start your IoT journey](#)

### More tutorials

You can find more tutorials in the [Arduino IoT Cloud documentation page](#).

