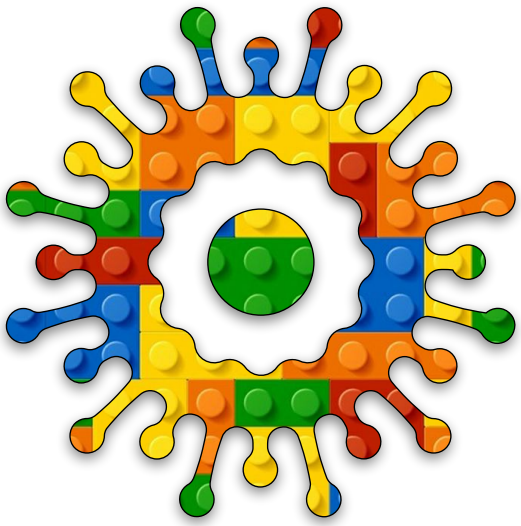
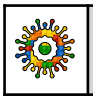
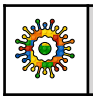
Internet of Things Module B Unit #1 your first Thing





Contents



Setting up the software

There are a number of steps I will go through with you to get started. I use a MacBook, but the principles should apply to Windows, Linux, and ChromeOS users because it is web-based. However, I did find out recently that you will need to use Chrome; other web browsers might work, but I had problems with Safari.

Also, because I have already used this many times, there may be extra steps you need to make that don't appear in the tutorial. All I am going to do is give you a broad outline of the process of getting set up.

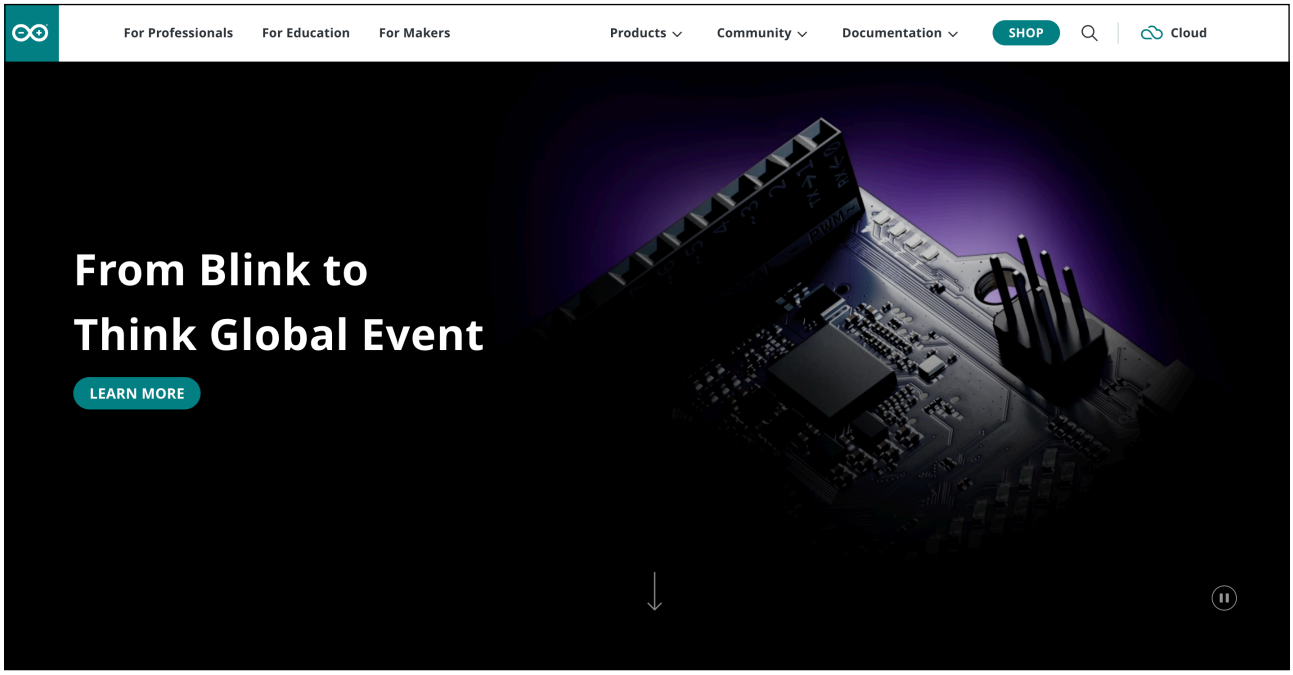
Assuming you have a device, then you can plug it into your computer. The steps below are just a very simple list, and I will cover these in more detail. There is a lot to cover, and I can't cover every little detail, but I will do my best. As we become more confident, then I can reduce the amount of detail.

- Is to set up an account with the Arduino Cloud.
- Once you are logged in, you need to add the device.
- It will ask you to install the Arduino Cloud Agent.
- You will need to give it a name.
- You will need to download and keep safe the secret key and the device ID.
- Add the details of the Wi-Fi (including password) and the secret key to the device.
- We create a Thing and give it a name.
- We link the (associated) device to the Thing.
- We create the variables we want and name them.
- We add in the code that we will upload to the device.
- Check that we have all the device and Wi-Fi details.
- Upload the sketch to the device.
- You then create a Dashboard with widgets connected with that variable.

Getting started

When you go to the Arduino website https://www.arduino.cc, you will get this (or a similar) page. The next step is to click on the Cloud icon (top right fig.1).

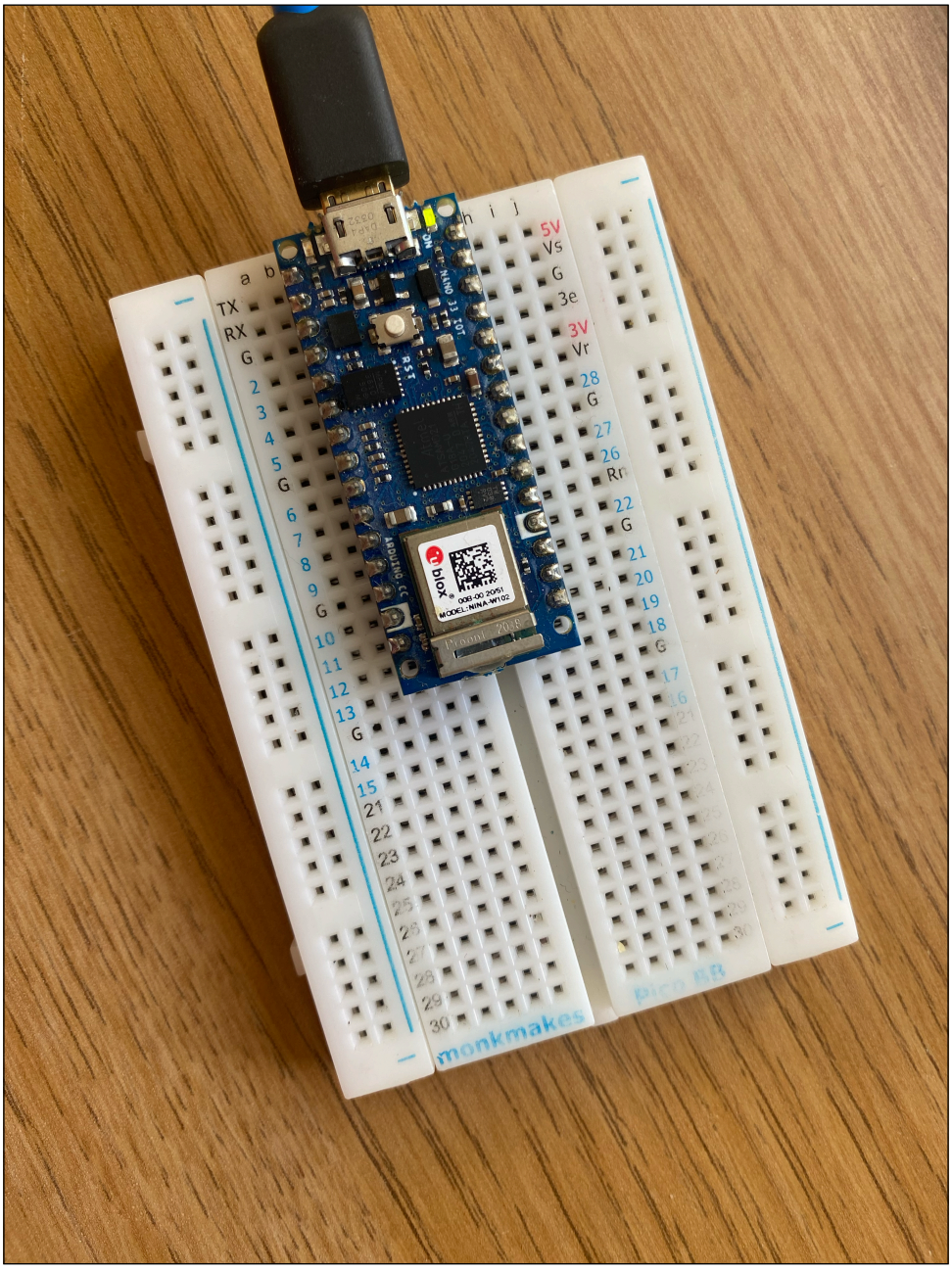
Figure 1: click on the the Cloud



Connecting the device

Plug in your Arduino Nano 33 IoT device to your computer via the USB cable. You should get a green light LED, which tells you that there is power to the device.

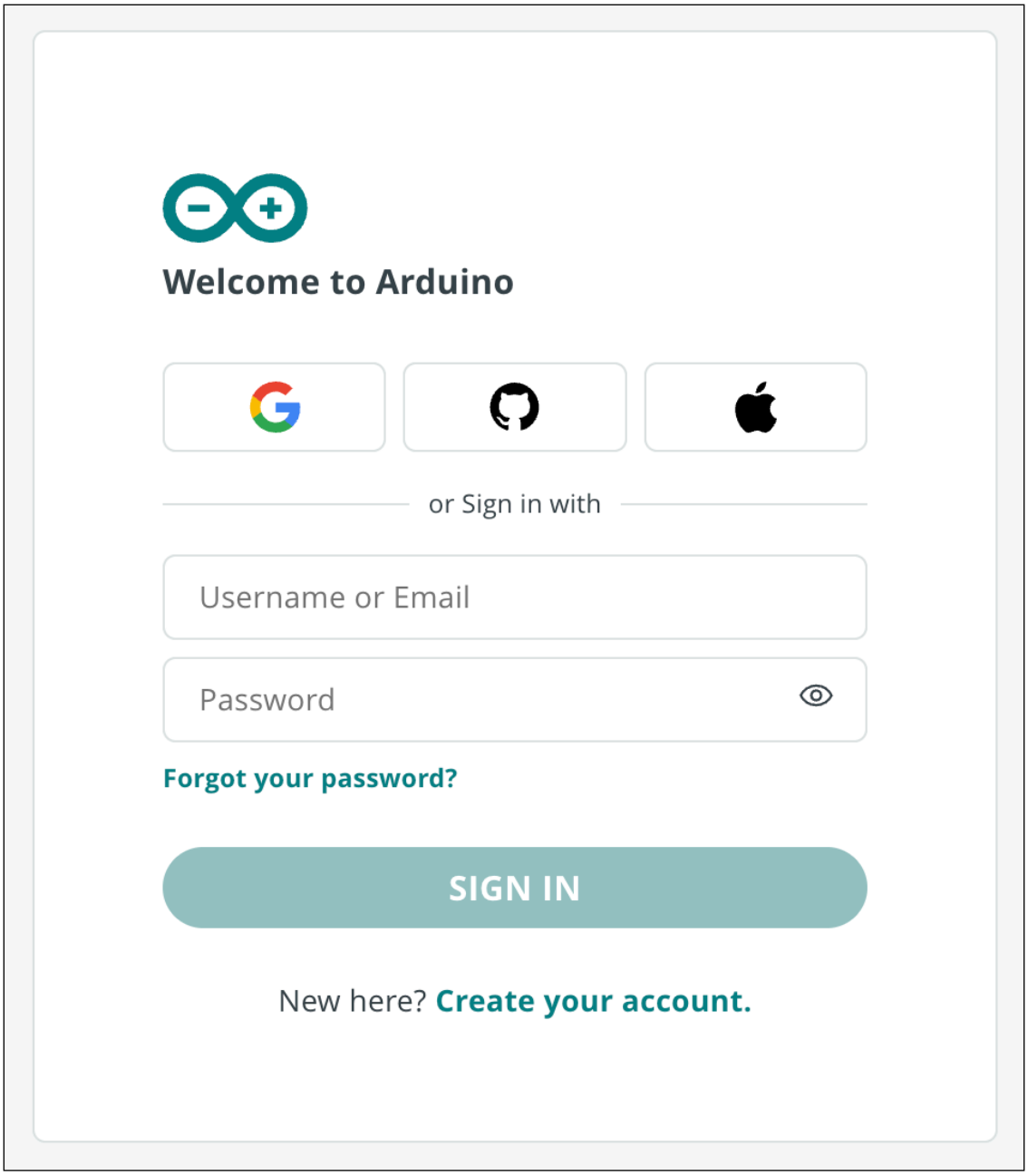
Figure 2: connecting the device



Sign up

You will need to sign up to use the cloud feature.

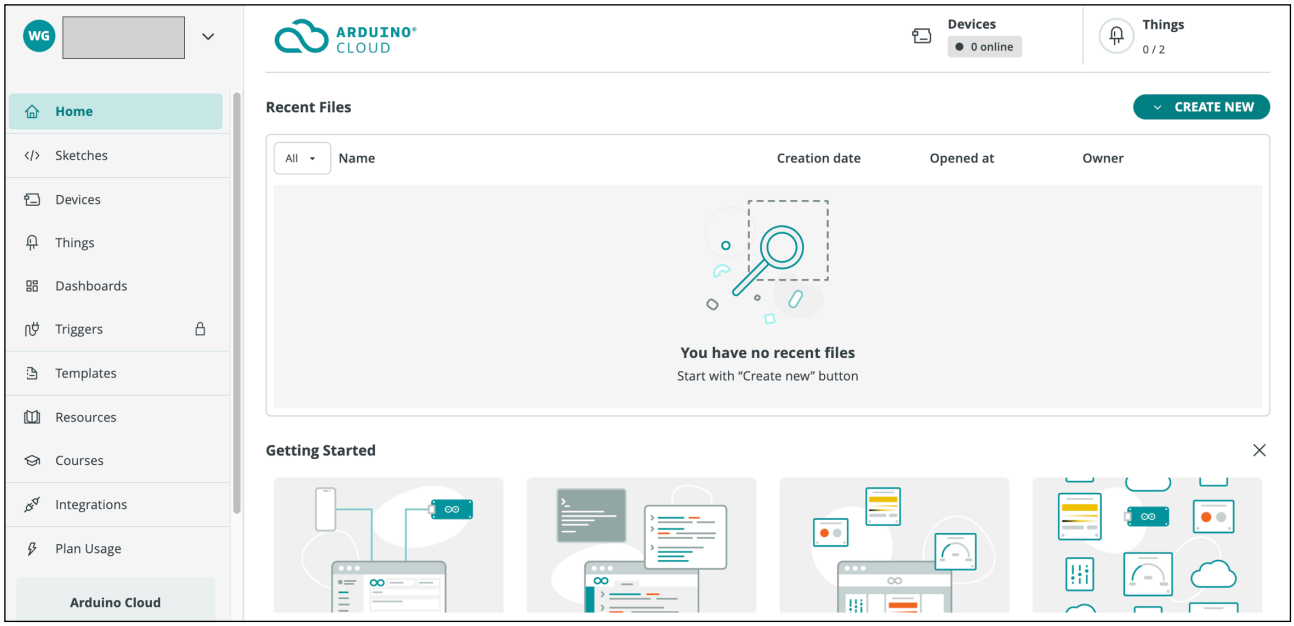
Figure 3: sign up



Home Page

Once you have signed in, you will get a home page like fig. 4. You will see a list on the left-hand side and two information tabs at the top right-hand side.

Figure 4: home page

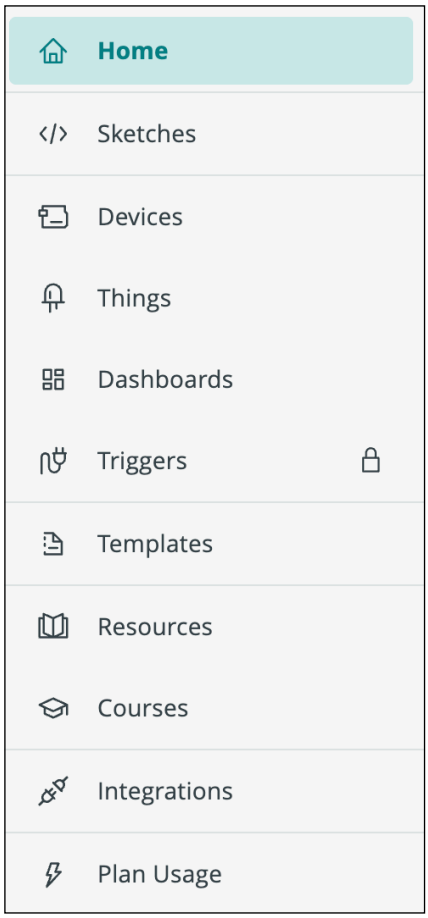


Menu list

This is the panel we will be using a lot to navigate the website. One thing to point out is that we are using the free version, which has some limitations. We can only upload 25 times, we are limited to the number of Things (two) we can create, and the number of devices we can add.

For this tutorial, this will be fine, but if you want more features, then you need to upgrade to a plan, which will give much more access if you are serious about developing projects (which I hope you are). We may well look at some of the other features later on with a paid plan, but for now, let's start with the free version.

Figure 5: menu list



Devices and Things

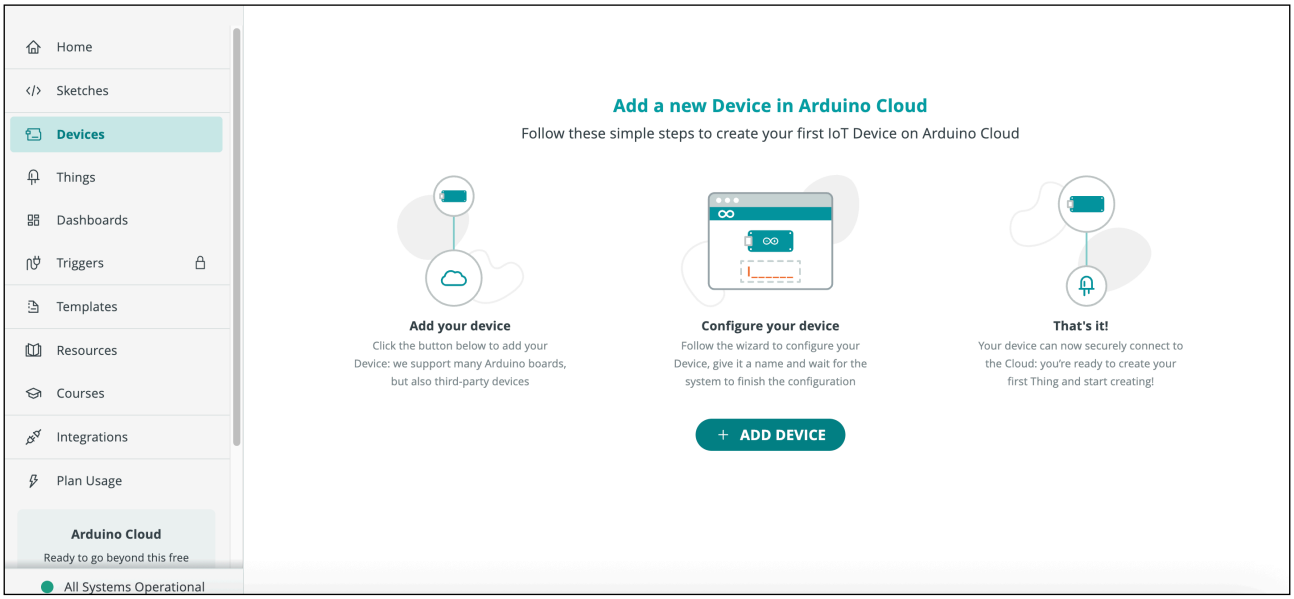
You can see that we have no `Devices` and no `Things`. We first need to add a device. Assuming you have an Arduino Nano 33 IoT device, plug it into your computer. Next, we need to add a device.

Figure 6: Devices and Things

Devices

If you click on the Devices tab on the left, you will get a page such as this fig.7 and we want to ADD DEVICE.

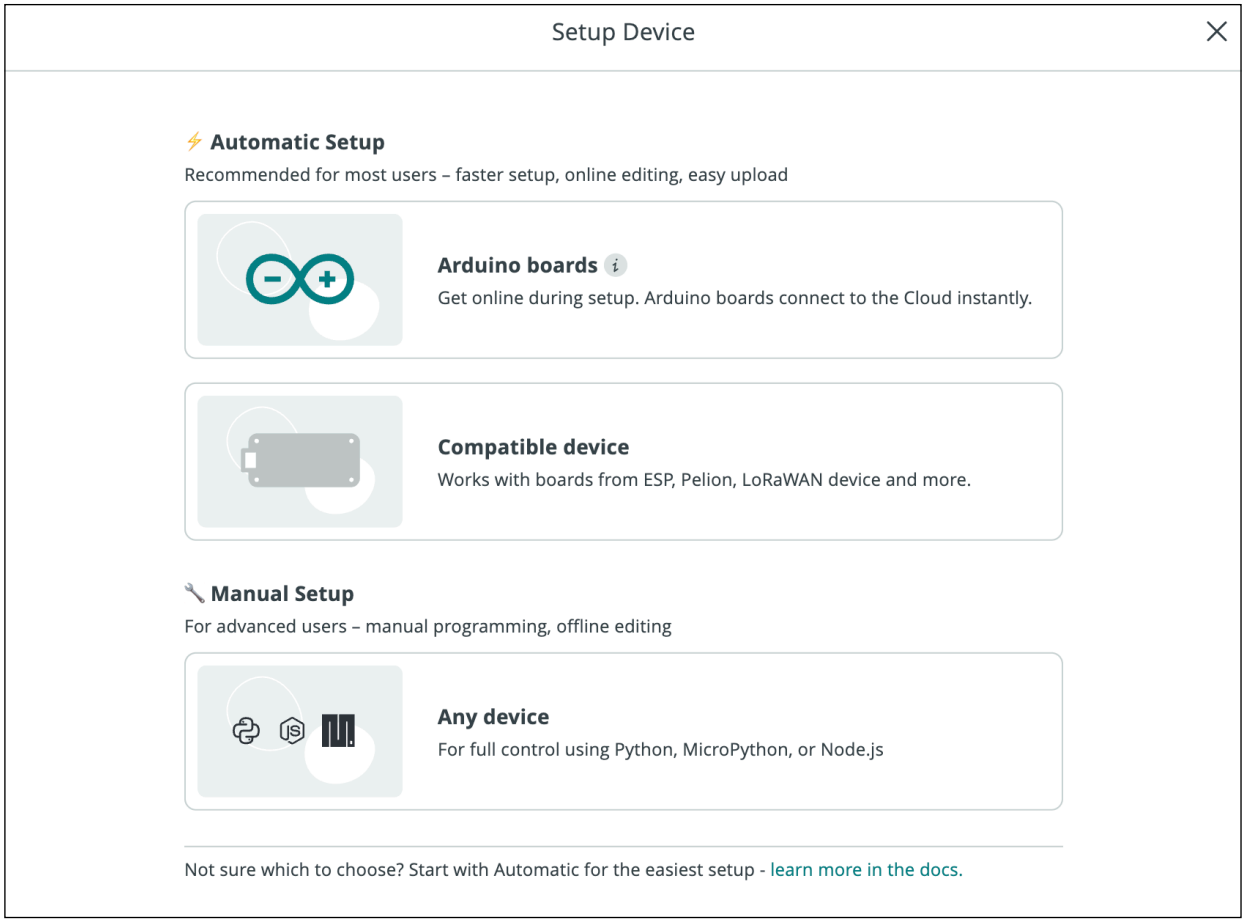
Figure 7: Devices



Adding the device

This bit can get a bit complex, so if at first you don't succeed, just keep on trying. We have an Arduino Board, so you can click on the top section button. If in the future you are using non-Arduino boards such as ESP boards, then you will have to check if it is compatible and click on the button below it. The manual setup is as it says for more seasoned users.

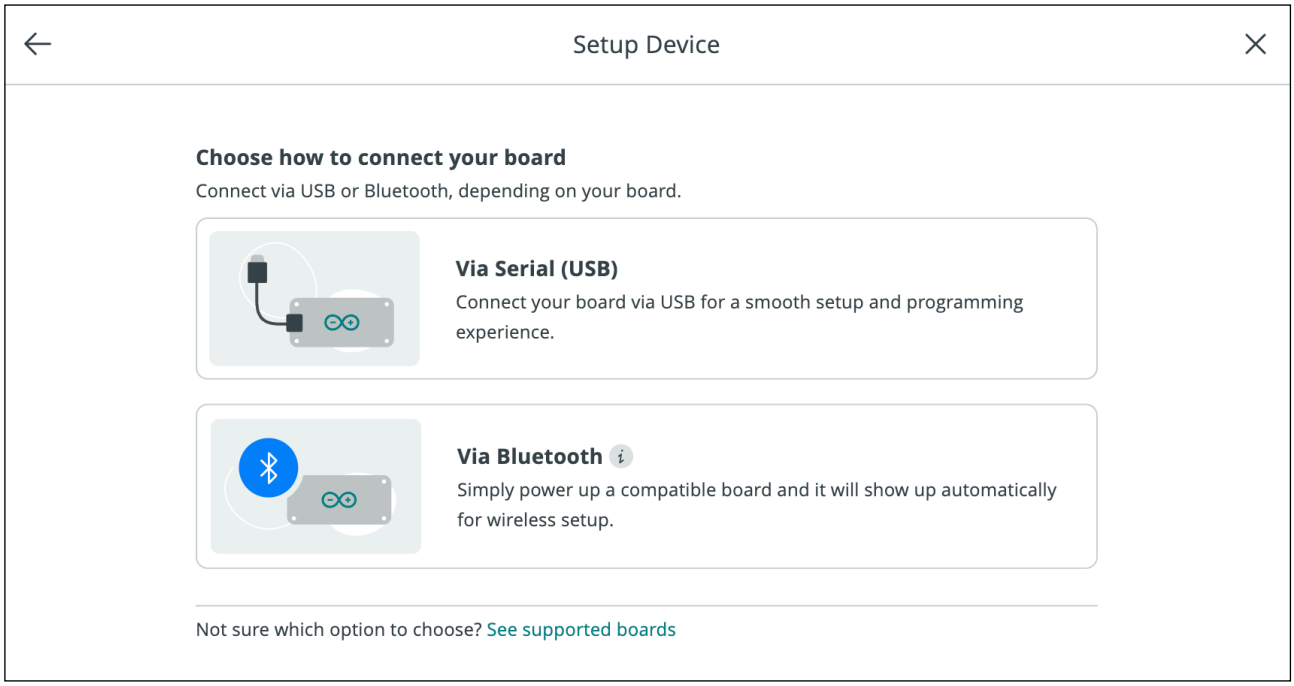
Figure 8: Arduino boards



USB setup

You can add the device two ways, through USB or by Bluetooth. I recommend USB, but if you are being brave, then try the Bluetooth approach. Click on the top button, making sure your device is connected by USB and the little green light is on.

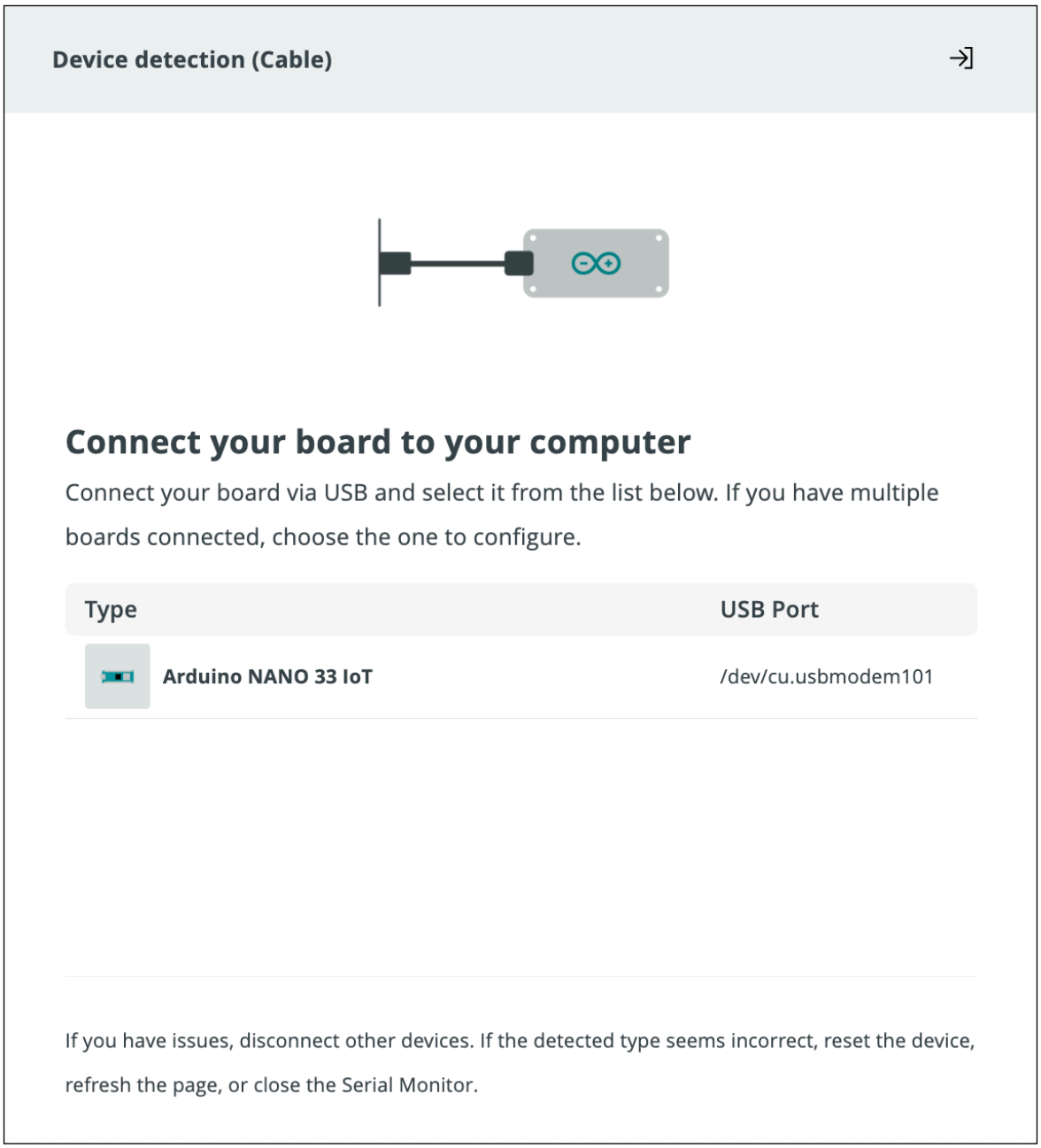
Figure 9: USB connection

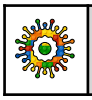


Detecting the device

Connect your device if you haven't already done so. It is going to search for it. It should pop up and show which port it is connected to. As you can see, you could do several at once. Once you can see it, then click on it to select.

Figure 10: detecting the device





Installing the cloud agent

I already have it installed. It is at this point that it will ask you to INSTALL the Cloud Agent. It will say it is missing. You will only have to do this once. This allows the website to connect to the device on the port. It won't work if it isn't installed. Depending on the machine you are working on, the steps to take will depend on the machine. So I won't cover them here because they can vary.

If you have problems, then I suggest keeping trying. Sometimes it takes a few goes to get it to work, but generally with Chrome, it works straight away. If you are using a Chromebook, it may well ask and then skip the process.

You should get an icon to show you it is installed and running. If it is greyed out, then click on it and resume it.

Figure 11: the message you will get

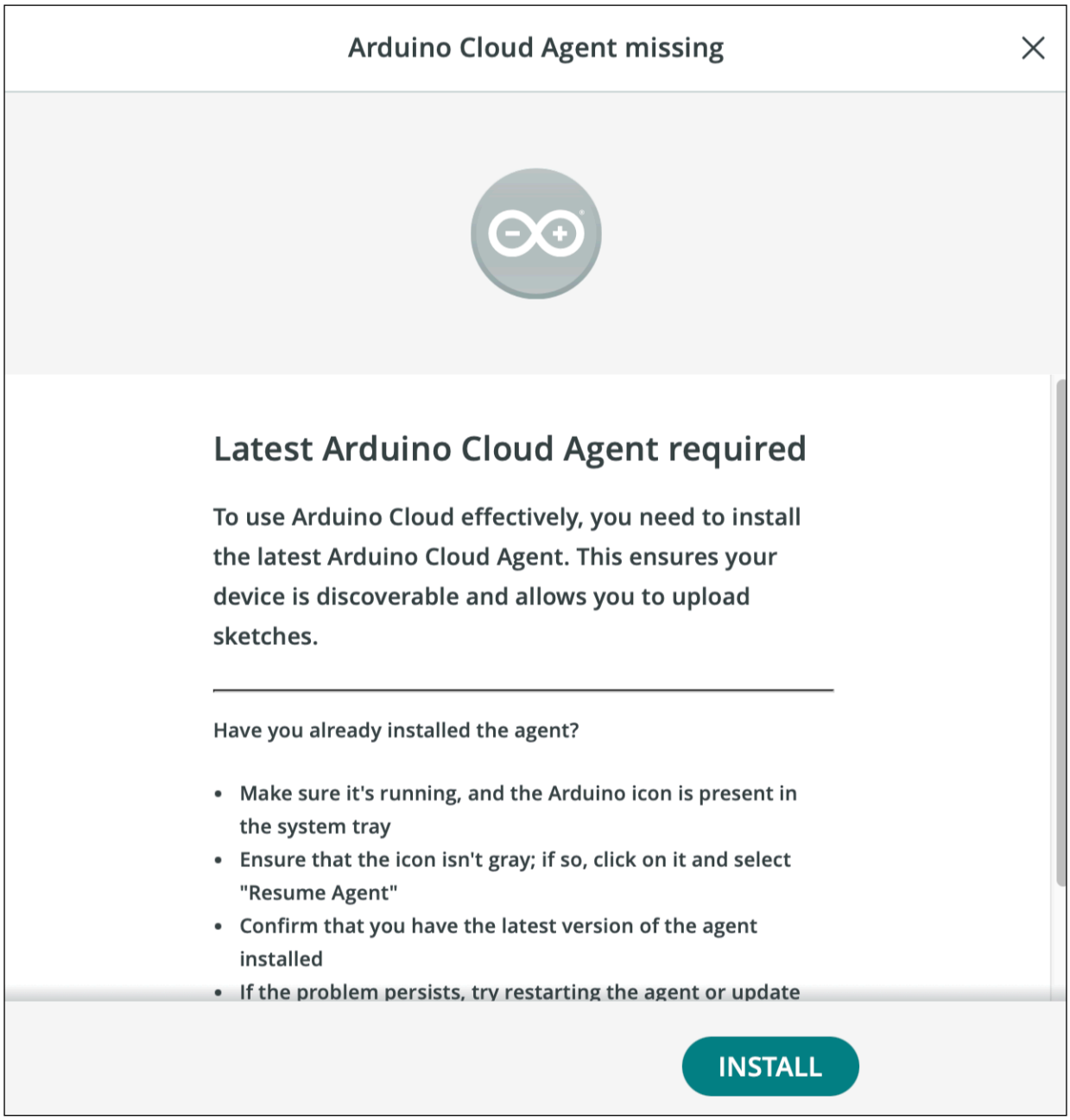
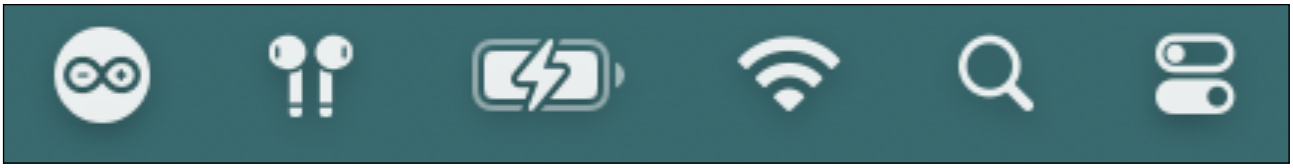


Figure 12: cloud agent icon

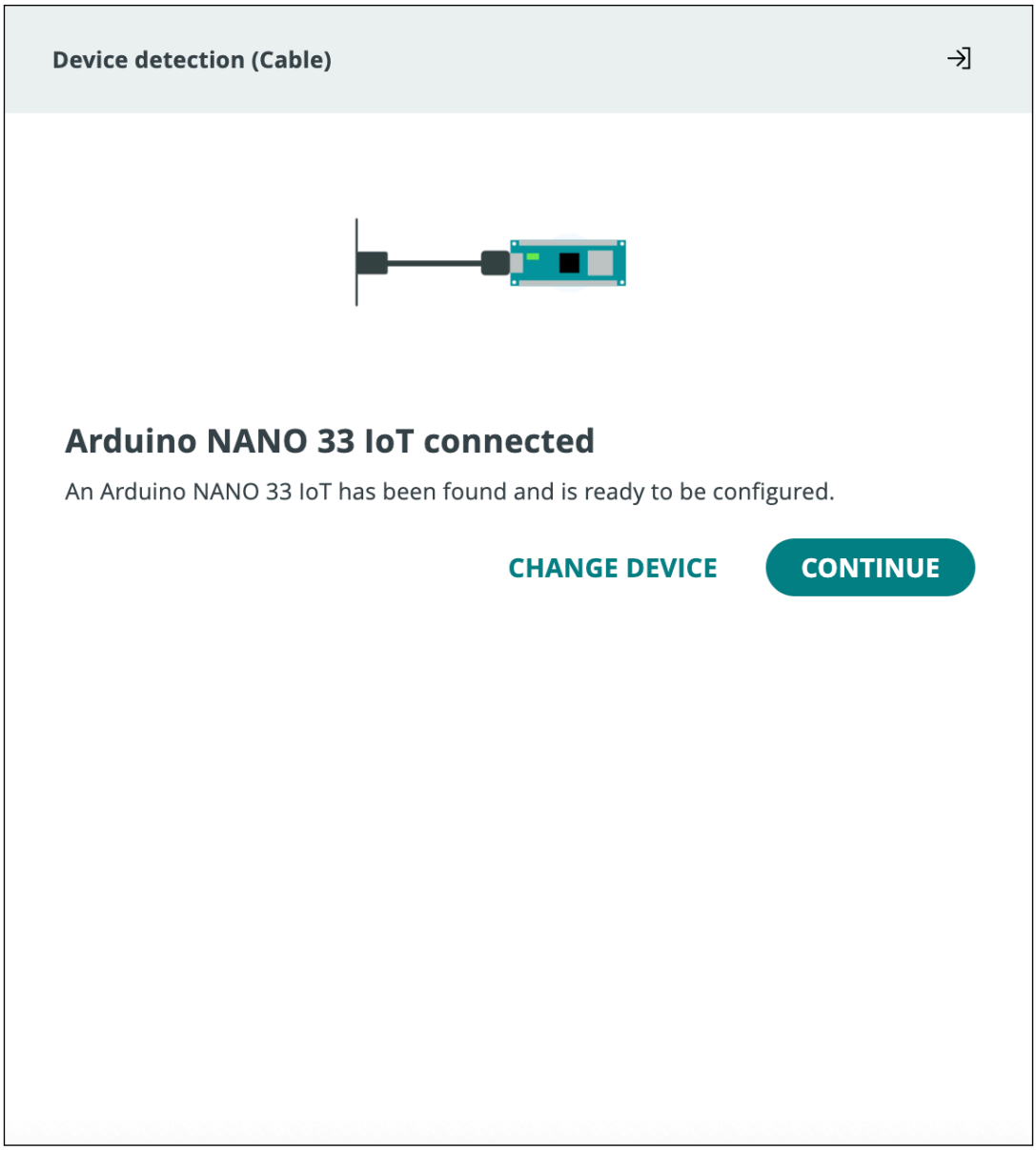


Give it a name

Once it has discovered the device, click on CONTINUE.

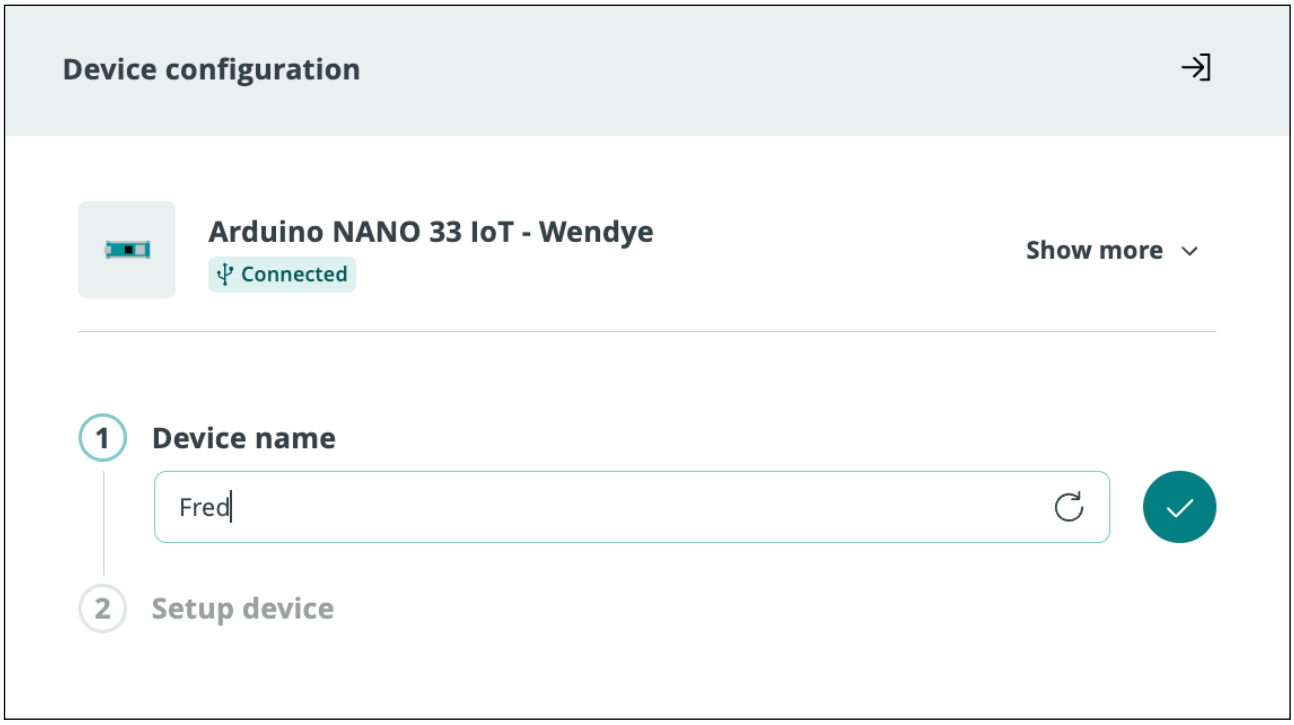
It will suggest a name for your device. I like to give it a name of my choosing, so I am going to call it Fred.

Figure 13: detected



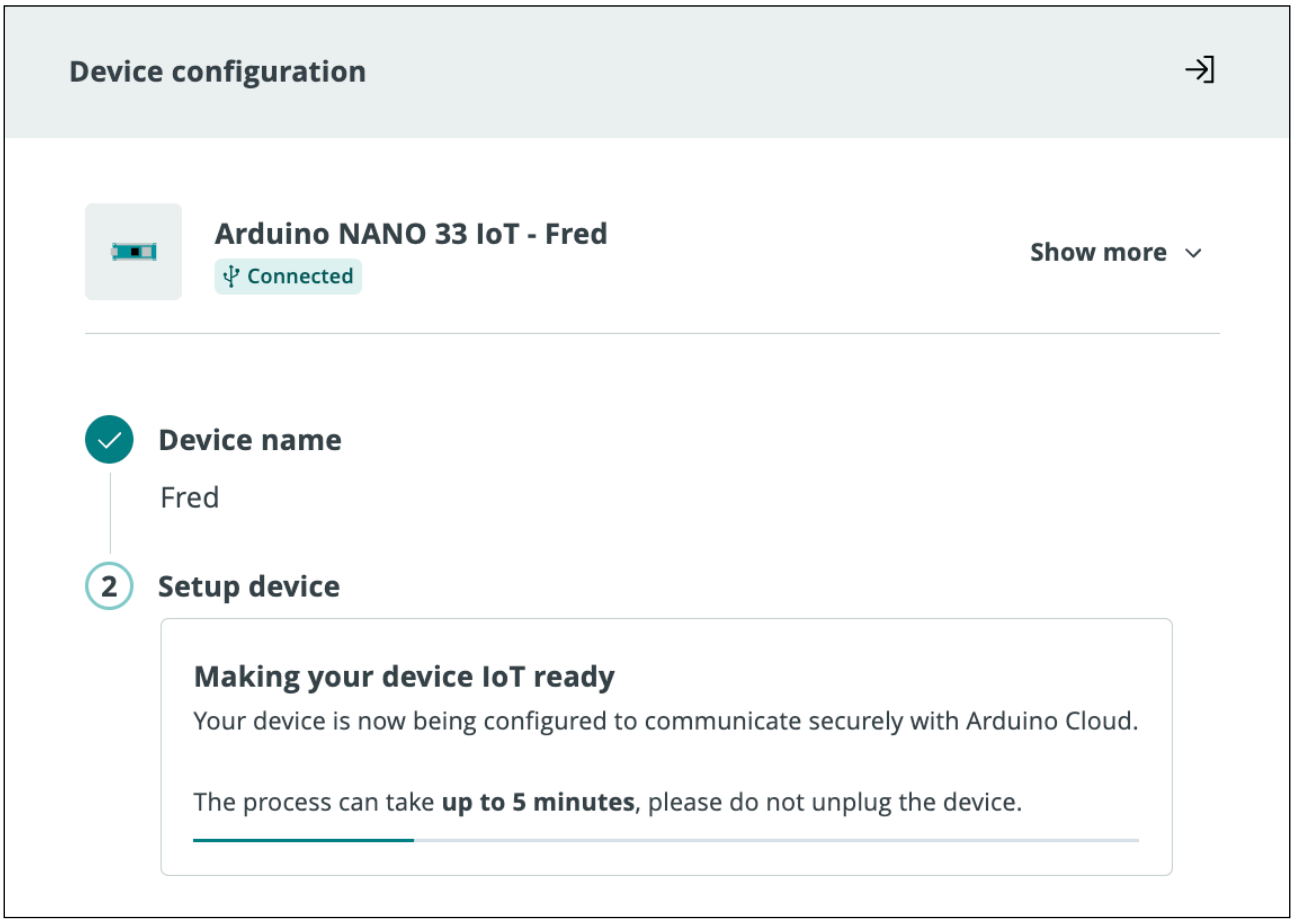
Once you have given it a name (or not), then click on the tick circle.

Figure 14: configuration



Waiting for the device to be set up as an IoT device. This is often where I have had problems. If it gives you an error message, it is worth checking to see if it has actually done it; otherwise, keep on trying.

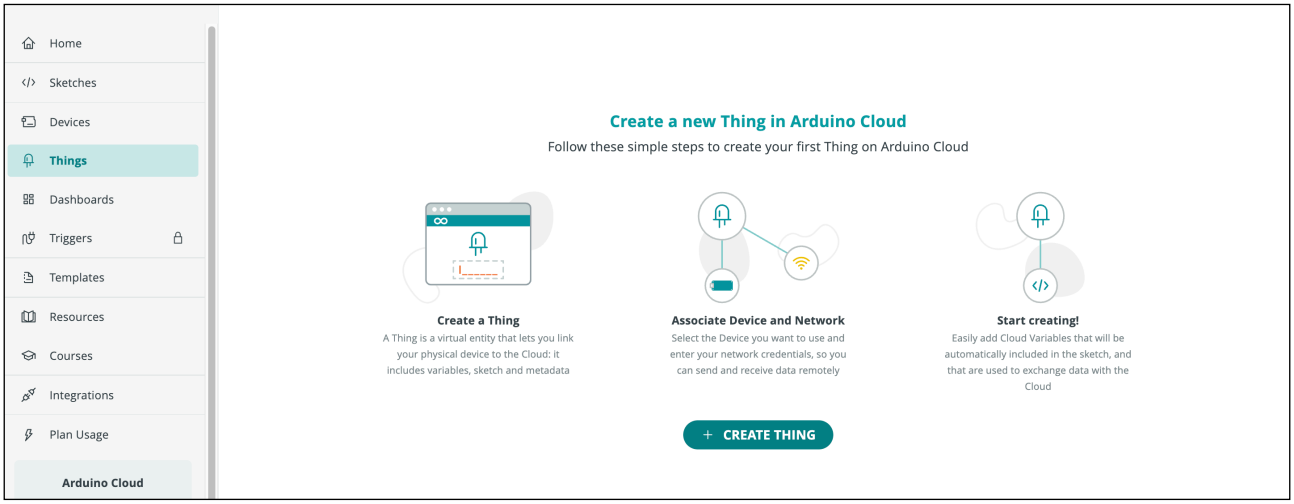
Figure 15: a waiting game



Creating a Thing

Hopefully yours went through OK. Next, head over to Devices to see your device; it will be offline. For the next step, we need to create a Thing. In the Things tab, click on the CREATE THING button.

Figure 16: the Things tab

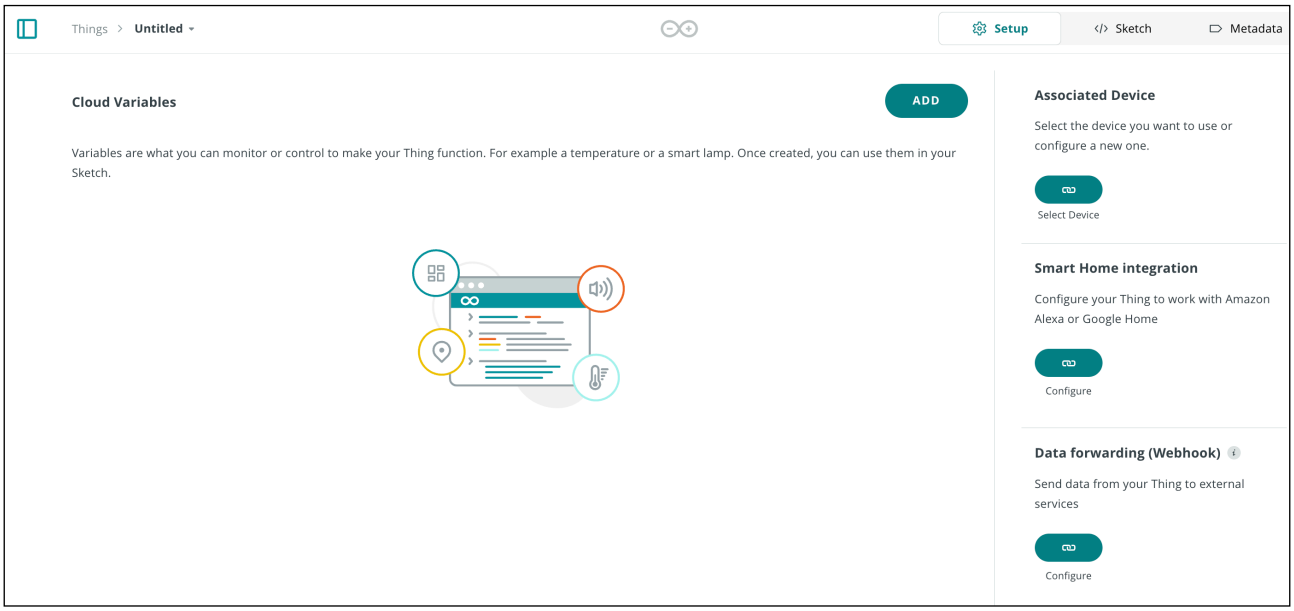


Building the Thing

This is where we have to make a number of decisions. A Thing is where we create the IoT environment for the device and what it is going to do. There are a number of aspects which we will look at in turn:

- Give it a name (currently untitled)
- Add a variable (cloud)
- Add an associated device (your Arduino Nano 33 IoT device)

Figure 17: our Thing



Renaming the Thing

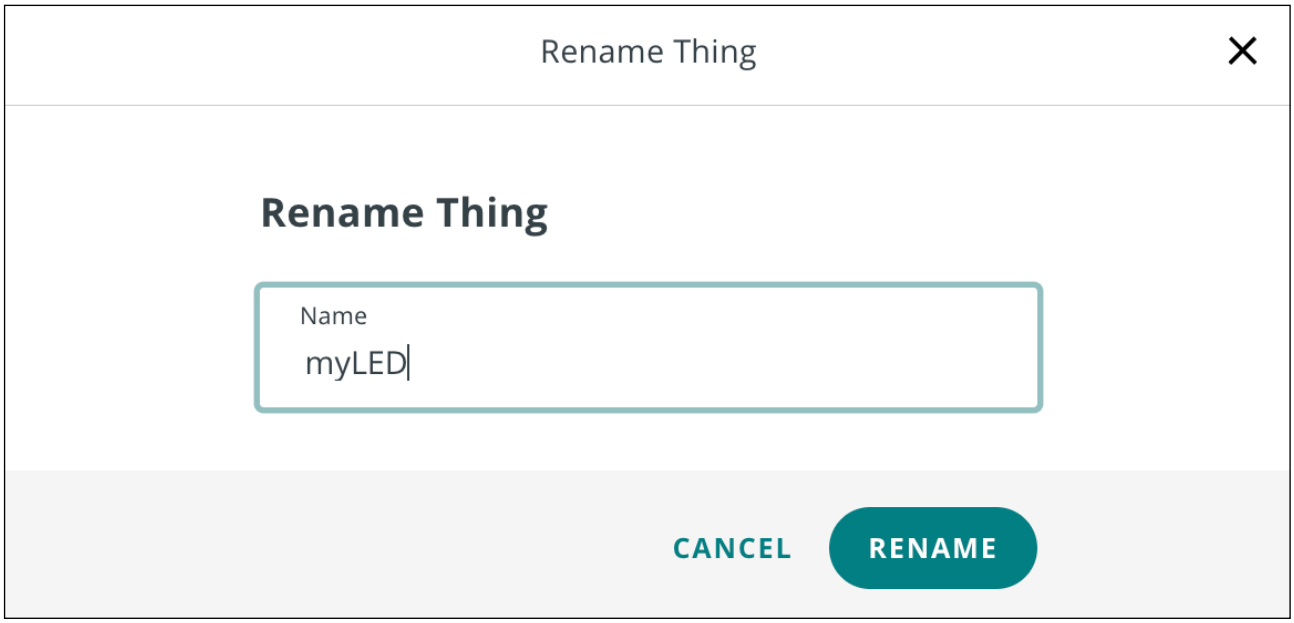
We haven't talked about what we are actually going to do with the device. This is important now because we are going to give all this context. What we are going to do is switch the built-in LED on and off using the power of the internet. We have a power LED which you would've seen already, on the other side of the board (near to the power LED) is another LED, which we can use. It is called a `BUILT_IN LED` and it is associated with pin `D13`.

As an option (you can just use the built-in LED) I added the LED traffic lights to the circuit (pin D2), so that will also be switched on/off at the same time as the built-in LED.

We need to give a name to this thing. We can call it anything, but I suggest we call it `myLED` for now. I can't think of a better name just at the moment. I don't want to call it `LED` because we are going to give the variable that name.

Click on the arrow next to the word `Untitled`. Change it and then click `RENAME`.

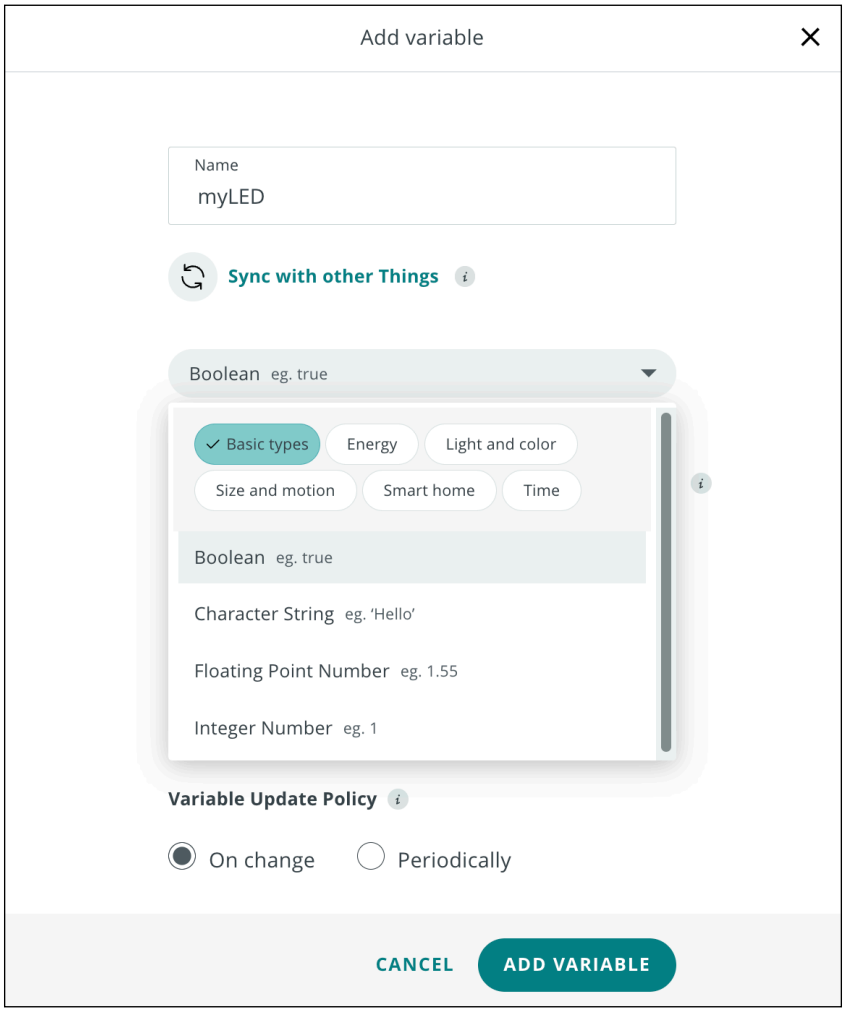
Figure 18: renaming the Thing



This takes you to a new information box; also, you can see the button with ADD written. This is where we add a cloud variable. What we want is what is called a boolean variable. This means it is either true or false, on or off. If you are unfamiliar, do not be concerned, you can run through module A units #3 to #7, they will go through all the basics if you are if need to.

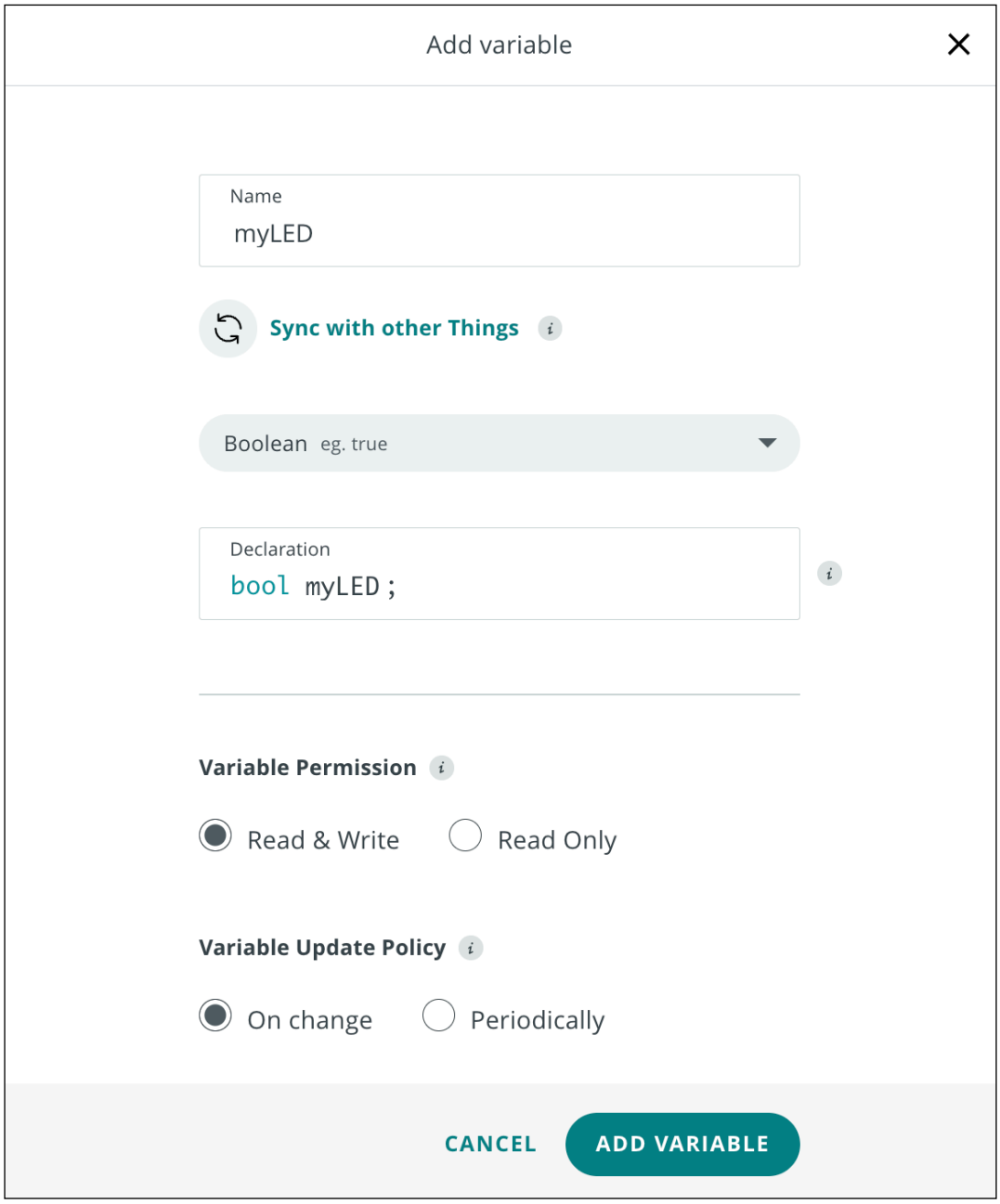
We have to give it a name (suggest: start with a lowercase letter), and the simple name is myLED. Then we need to select the variable type. Select Boolean eg. true. It will then create that variable with that name, bool myLED.

Figure 19: the Boolean option



We keep it `Read & Write`; everything should be the same. Now click `ADD VARIABLE`.

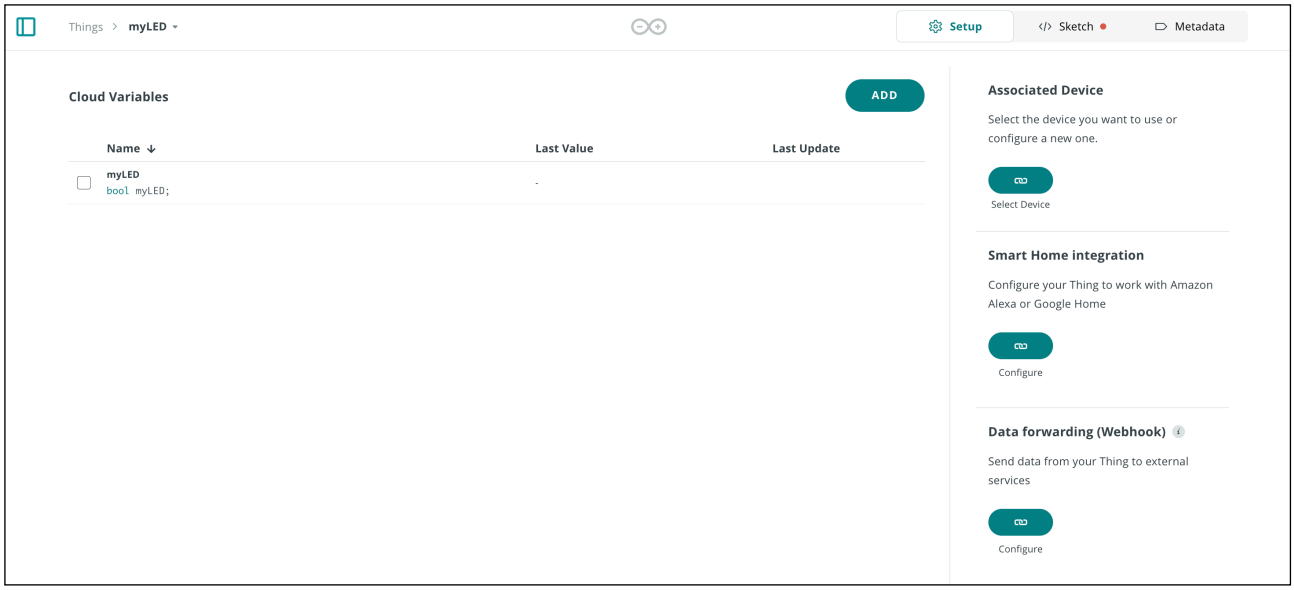
Figure 20: adding the variable



Selecting the device

We have a cloud variable called `myLED` and it is a boolean (`bool`) data type. You could add more variables by clicking on the `ADD` button, but we only need the one variable for now. Our next task is to connect the device (our `Fred`) to the `myLED` Thing. Click on the `Select Device` button under `Associated Devices`.

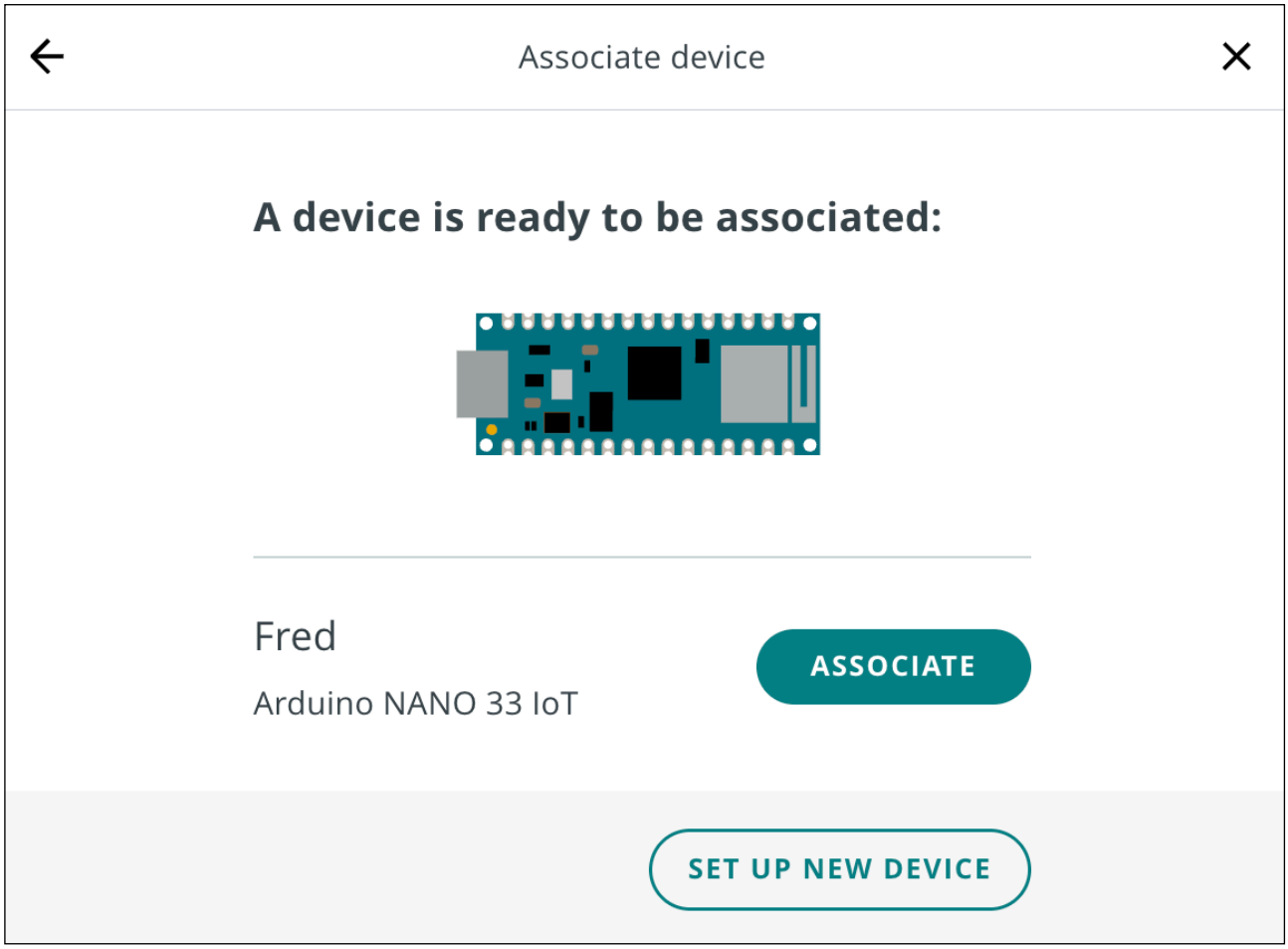
Figure 21



Associated device

This gives us a chance to check we are connecting the right device to the right Thing. This is because you might end up with many devices or more than one device connected. If it is OK, then click on the ASSOCIATE button.

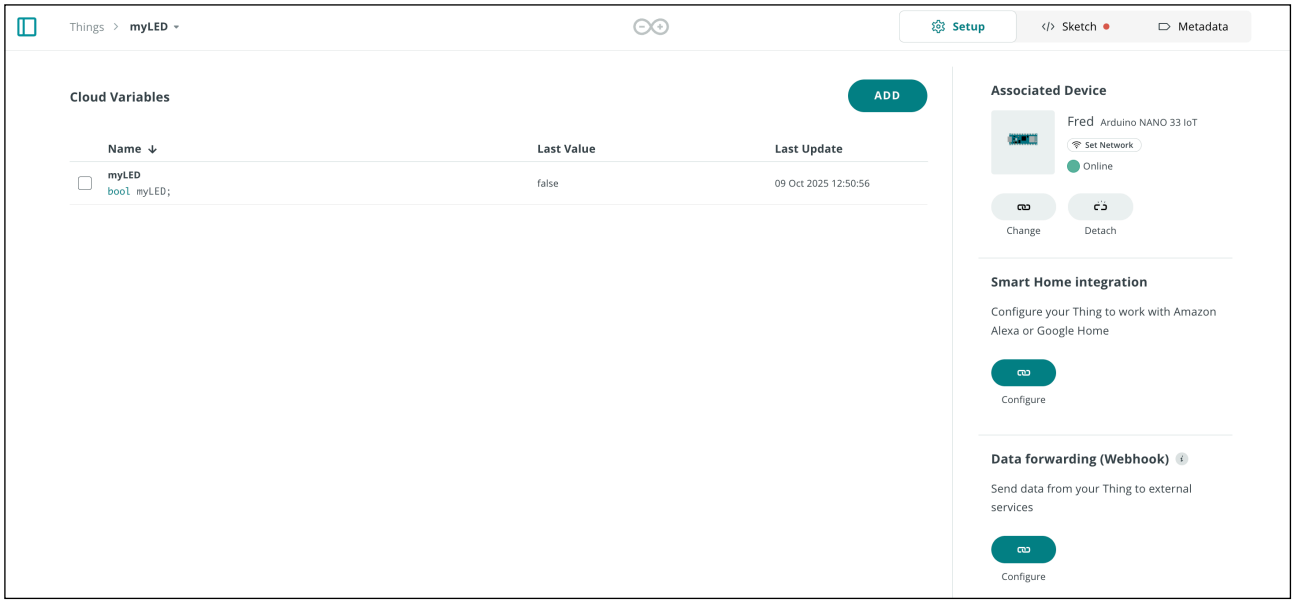
Figure 22



Device associated

We are well on our way. You should see the device is now associated with the myLED Thing. Our next task is to set the network. For this, we need to go back to the device and select our Fred. The menu has collapsed, so click on the box-like icon in the top left, and the menu will reappear with the contents. Click on devices and then click on your device; in my case, it is Fred.

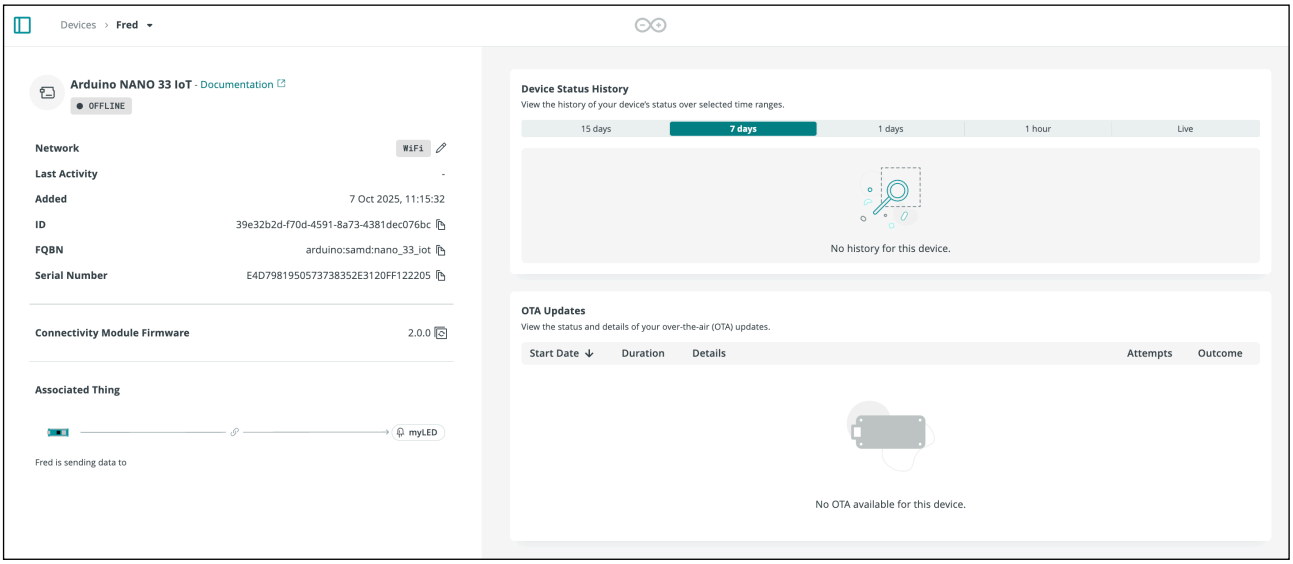
Figure 23: device associated



Setting the network

For this, you will need the name and password of your Wi-Fi. On the page for your device, you will see a lot of information. We want to add the Wi-Fi credentials to the device. The device needs to have them in the code itself so that it can connect to the Wi-Fi wherever it is. Click on the grey box with the word `Wi-Fi` inside it opposite `Network`; it will bring up another box to fill in your Wi-Fi details.

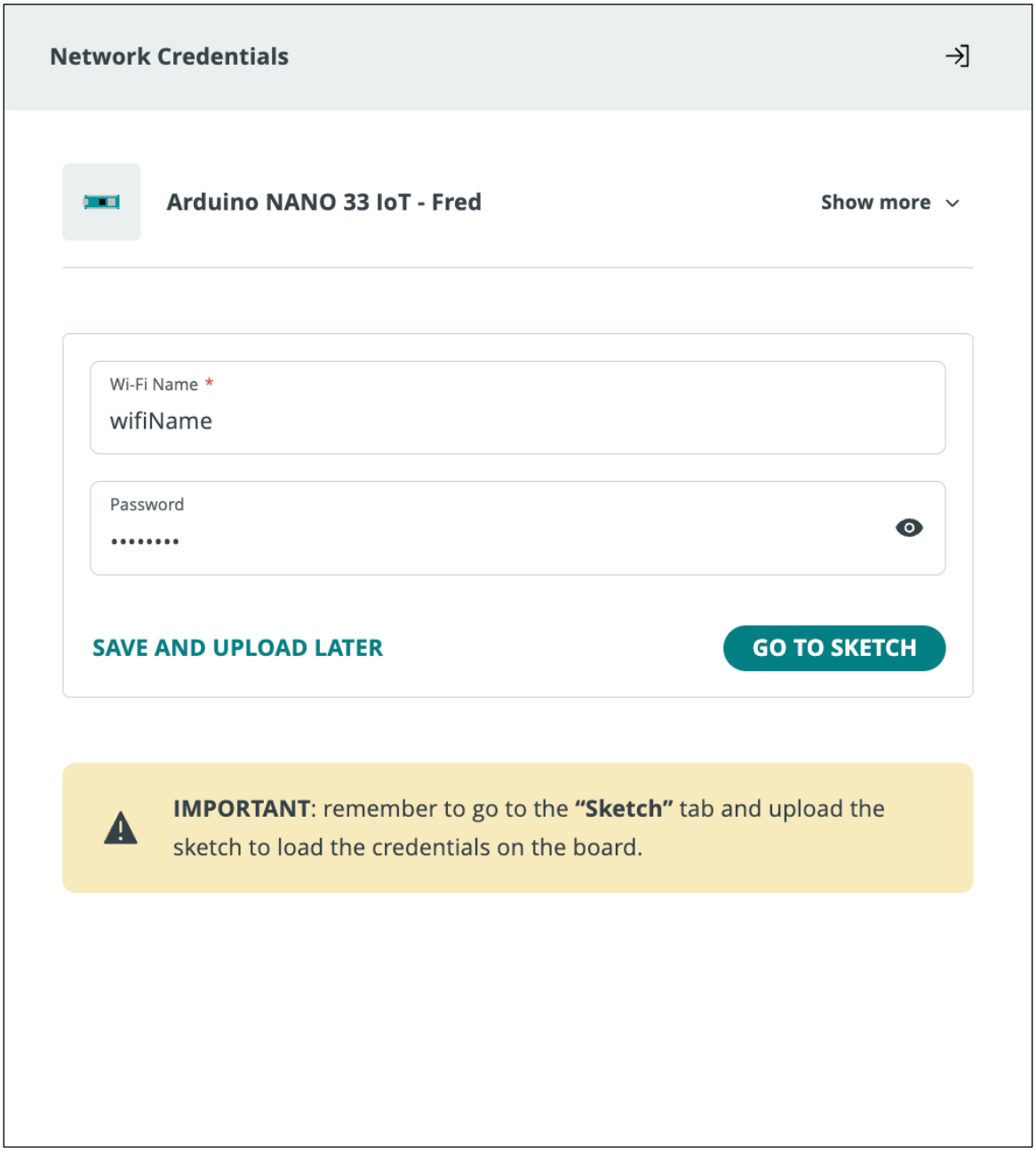
Figure 24: WiFi credentials



Filling the details

Put in the details and then you will need to go to the sketch, which is the business end of things. Now click on the GO TO SKETCH button.

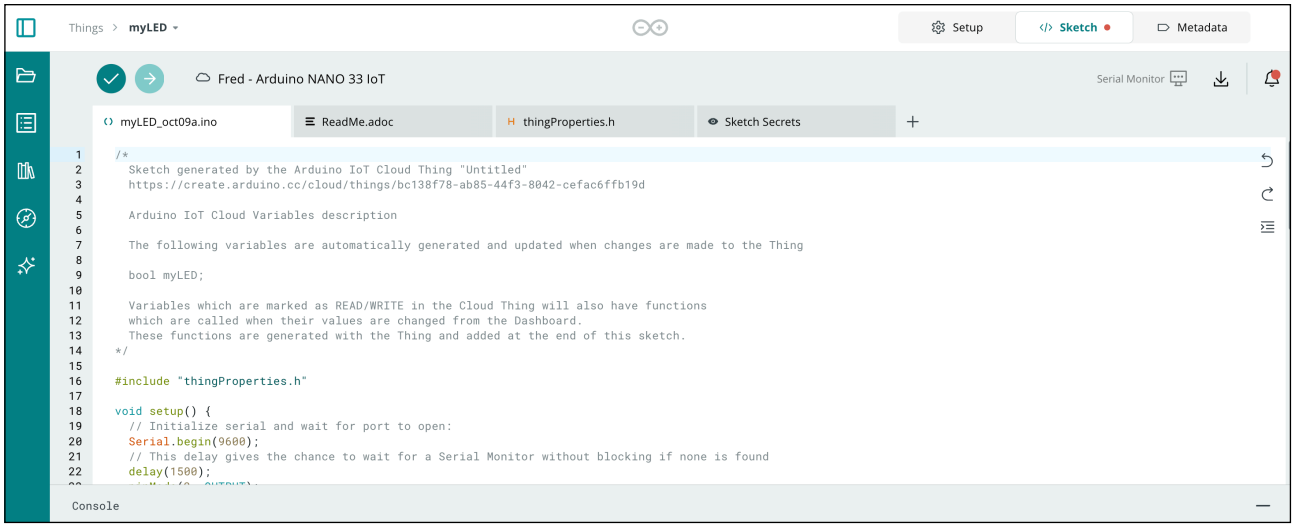
Figure 25: filling the details

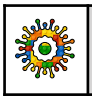


The sketch

This is the code that is generated. If you are familiar with coding the Arduino, then you can see all the additional comments, which are worth a read. If you are new to coding, then you will need to follow the steps, but for now, I will go through what each one of the four tabs means.

Figure 26: the sketch





The main sketch (first tab)

The first tab contains nearly all the information you need. We will have to add the code to make it work, but we will come to that in a moment. I have greyed out the comments; we can remove them shortly and add in our own code.

❗ don't delete anything that isn't a comment.

```
/*
  Sketch generated by the Arduino IoT Cloud Thing "Untitled"
  https://create.arduino.cc/cloud/things/xxx-xxxx-xxx

  Arduino IoT Cloud Variables description

  The following variables are automatically generated and updated
when changes are made to the Thing

  bool myLED;

  Variables which are marked as READ/WRITE in the Cloud Thing
will also have functions
  which are called when their values are changed from the
Dashboard.
  These functions are generated with the Thing and added at the
end of this sketch.
*/

#include "thingProperties.h"

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
```

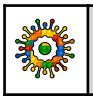
```
  // This delay gives the chance to wait for a Serial Monitor
without blocking if none is found
  delay(1500);
  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  /*
     The following function allows you to obtain more information
     related to the state of network and IoT Cloud connection and
errors
     the higher number the more granular information you'll get.
     The default is 0 (only errors).
     Maximum is 4
 */
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
  // Your code here



}

/*
  Since MyLED is READ_WRITE variable, onMyLEDChange() is
  executed every time a new value is received from IoT Cloud.
*/
```

```
void onMyLEDChange()  {
  // Add your code here to act upon MyLED change
}
```

🗒️ Notes

There looks like quite a bit to unpack. The main thing is the variable name `myLED` and that it is identified as a boolean datatype (`bool`). The `void setup()` function is where we will have set up our LED. We won't be putting anything in the `void loop()` function. The other important function is the void `onMyLEDChange()` function because this is where we decide what happens when the boolean variable (`myLED`) changes from `false` (off) to `true` (on).

All the other listed functions are beyond the scope of this introduction and there is nothing you need to do to them or with them; they have been generated by the Arduino Cloud IDE so it will work. Just ignore them for now but do not delete them!



ReadMe (second tab)

The next tab along is a ReadMe file. This is just general information you can give someone if you want to share it with them. For instance, components or anything you feel might be useful. Much of what you read here is made up.

```
:Author: xxxx
:Email:
:Date: 09/10/2025
:Revision: version#
:License: Public Domain

= Project: {Project}

Describe your project

== Step 1: Installation
Please describe the steps to install this project.

For example:

1. Open this file
2. Edit as you like
3. Release to the World!

== Step 2: Assemble the circuit

Assemble the circuit following the diagram layout.png attached
to the sketch

== Step 3: Load the code
```

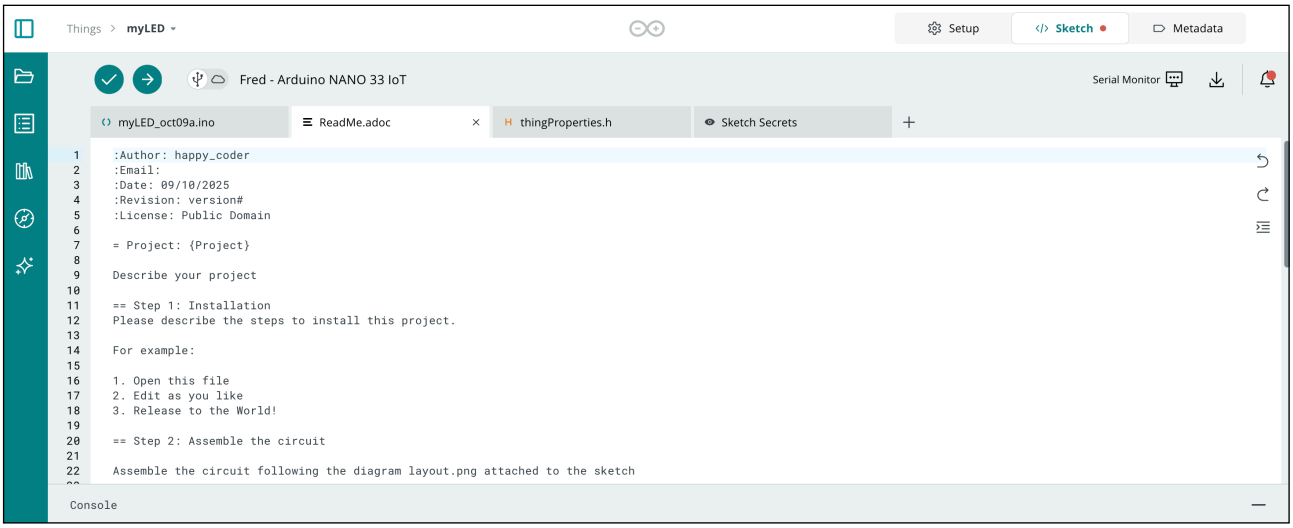
```
Upload the code contained in this sketch on to your board

=== Folder structure

....
 sketch123               => Arduino sketch folder
   ├── sketch123.ino     => main Arduino file
   ├── schematics.png    => (optional) an image of the required
schematics
   ├── layout.png        => (optional) an image of the layout
   └── ReadMe.adoc       => this file
....

=== License
This project is released under a {License} License.

=== Contributing
To contribute to this project please contact:

=== BOM
Add the bill of the materials you need for this project.

|===
| ID | Part name      | Part number | Quantity
| R1 | 10k Resistor   | 1234-abcd   | 10
| L1 | Red LED        | 2345-asdf   | 5
| A1 | Arduino Zero   | ABX00066    | 1
|===


=== Help
```

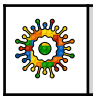
```
This document is written in the _AsciiDoc_ format, a markup
language to describe documents.

If you need help you can search the http://www.methods.co.nz/
asciidoc[AsciiDoc homepage]

or consult the http://powerman.name/doc/asciidoc[AsciiDoc
cheatsheet]
```

🗒 Notes

Feel free to delete, add or change anything, or do what I do and just leave it as is for now.

Figure 27: ReadMe





thingProperties (third tab)

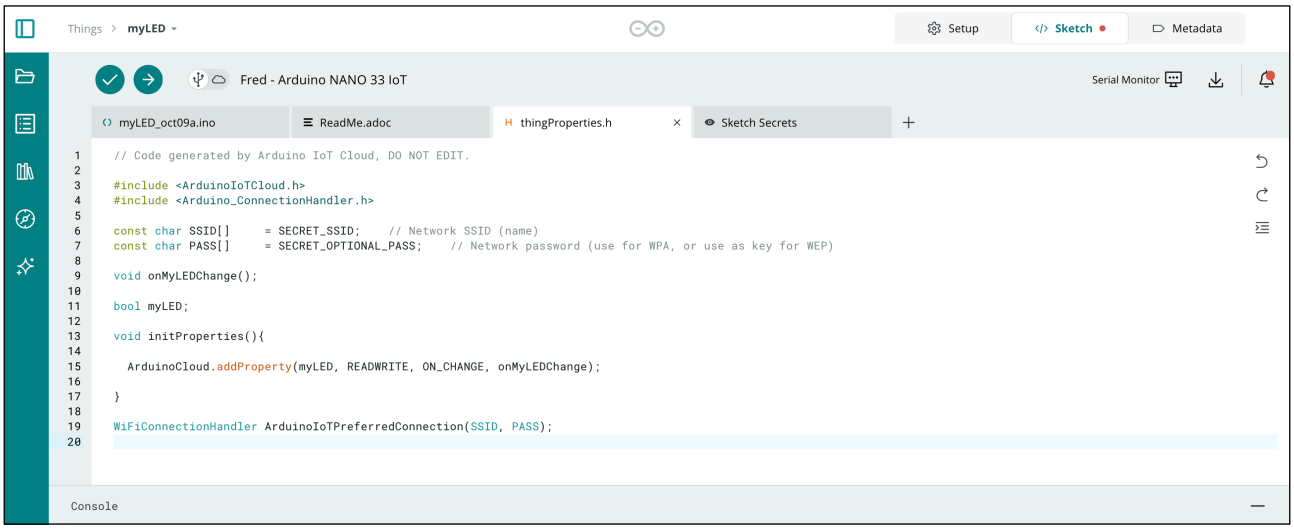
I probably don't need to say anything, but do NOT edit this page. This is important information that the main sketch will call upon. You should see some of the same functions as you get in the main sketch.

```
// Code generated by Arduino IoT Cloud, DO NOT EDIT.


#include <ArduinoIoTCloud.h>

#include <Arduino_ConnectionHandler.h>


const char SSID[]     = SECRET_SSID;    // Network SSID (name)

const char PASS[]     = SECRET_OPTIONAL_PASS;    // Network
password (use for WPA, or use as key for WEP)


void onMyLEDChange();


bool myLED;


void initProperties(){


  ArduinoCloud.addProperty(myLED, READWRITE, ON_CHANGE,
onMyLEDChange);


}


WiFiConnectionHandler ArduinoIoTPreferredConnection(SSID, PASS);
```

Notes

Nothing to change.

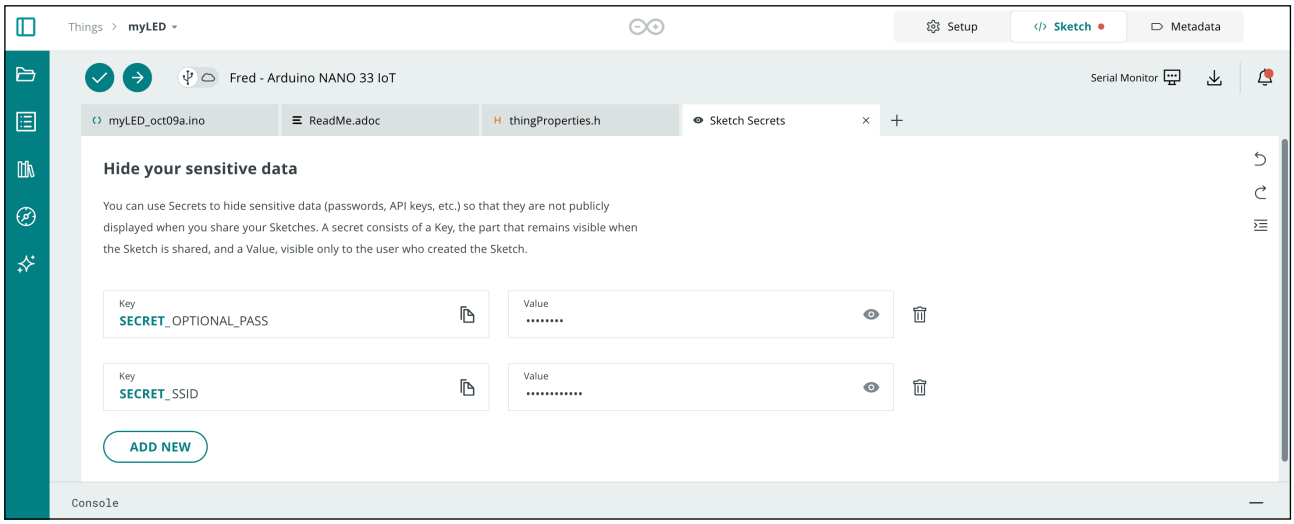
Figure 28: thingProperties.h tab



The secrets (fourth tab)

This tab holds the Wi-Fi name and password. You can add more secrets if you wish, like ID and secret key (but not essential).

Figure 29: secrets



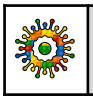
The main sketch

Back to the first tab and the main sketch. I want you to delete all the comments. This is just to make it clearer for when we add in the extra code we want. What you are left with is shown below.

```
#include "thingProperties.h"

void setup() {
  Serial.begin(9600);
  delay(1500);
  initProperties();
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
}

void onMyLEDChange()  {
}
```

Notes

This is the slimmed-down version of the code that is already there. It doesn't look so bad now.



Adding the code

This bit should be familiar to you if you followed the tutorial in module A. We are going to establish what we are going to do with each pin and what we want to see happen when we change the LED from off to on and vice versa. We don't have our switch just yet; we will come onto that in a bit.

We are going to use two LEDs, the built-in one and the traffic light LED on pin 2. You can choose to have both or either one.

```
#include "thingProperties.h"

void setup() {
  Serial.begin(9600);
  delay(1500);
  pinMode(2, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);
  initProperties();
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
}

void onMyLEDChange()  {
  if (myLED)
  {
    digitalWrite(2, HIGH);
    digitalWrite(LED_BUILTIN, HIGH);
  }
```

```
  else
  {
    digitalWrite(2, LOW);
    digitalWrite(LED_BUILTIN, LOW);
  }
}
```

🗒️ Notes

If `myLED` is `true`, it switches the LED(s) `on`; if `false`, it reverts to the `else` loop and switches it/them `off`.

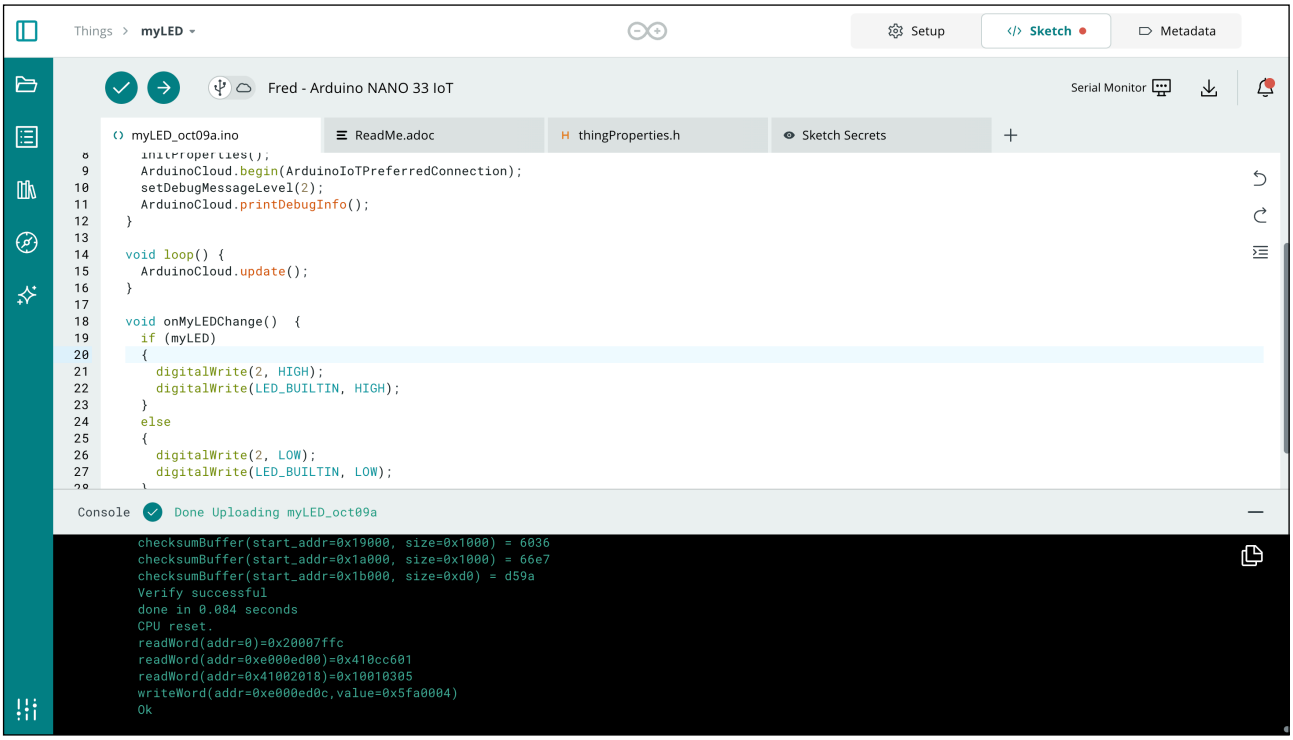
🛠️ Code Explanation

`if (myLED)`	This checks the boolean variable to see if it is true.

Upload the code

Once you have added the code to your sketch, you can upload it. Bear in mind nothing will happen just yet. You should get a nice green response in the console with a final Ok. If it is red, then you have a problem to solve. Make sure you are connected to your device and typed everything in correctly.

Figure 30: uploading sketch

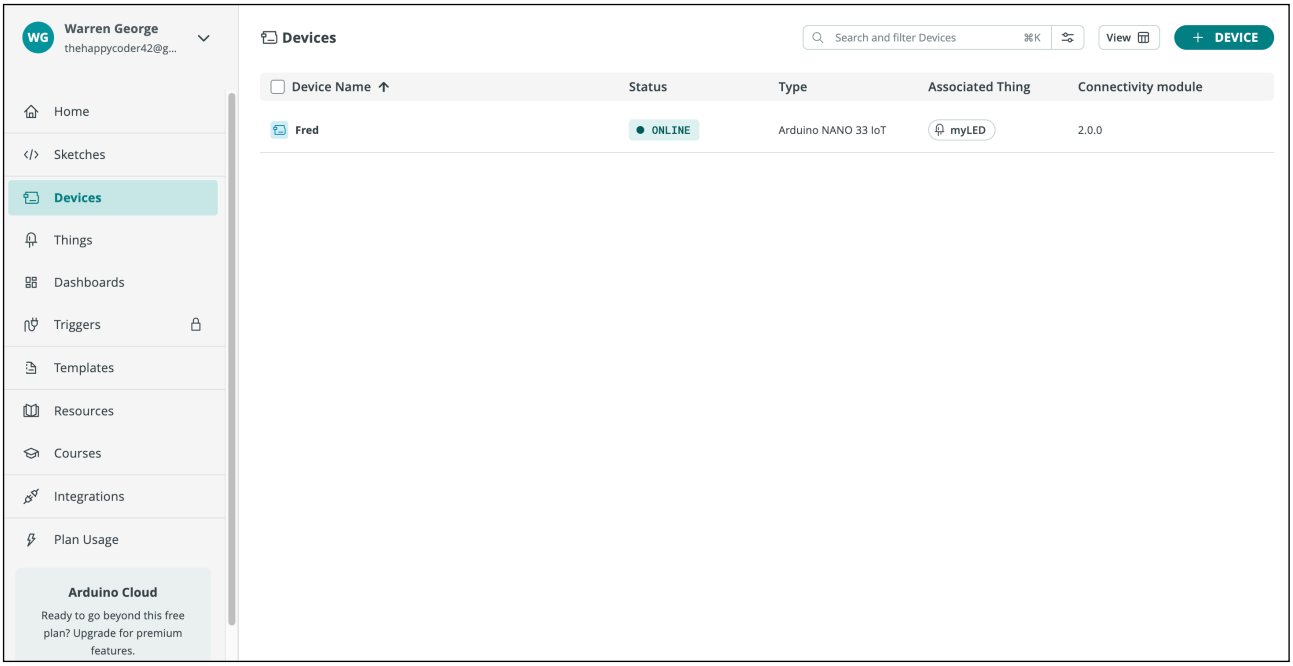


Online

Assuming everything uploaded OK. We can check to see if the device is online. Sometimes it may take a while to come online. There may be other issues. See troubleshooting section at the end of this tutorial.

Go to the Devices tab on the side menu, it should look something like the figure below. If not, then either just wait a bit longer or check you have the right WiFi credentials in the Secrets tab when you upload. For other solutions, see the troubleshooting at the end of the unit.

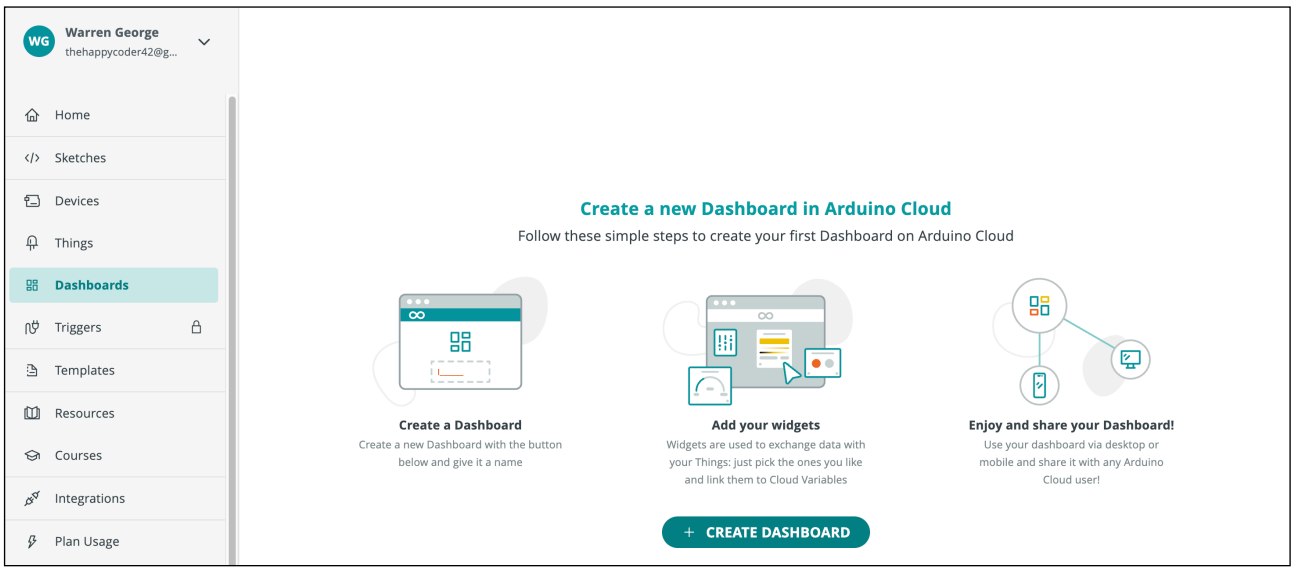
Figure 21: online (or not)



Creating a Dashboard

Assuming you are online. We need a switch to switch it on and off. For that, we need to go to the Dashboard tab on the menu side menu. Click on that, and you should have a page like the one below. You will not be surprised that we are going to click on the + CREATE DASHBOARD button.

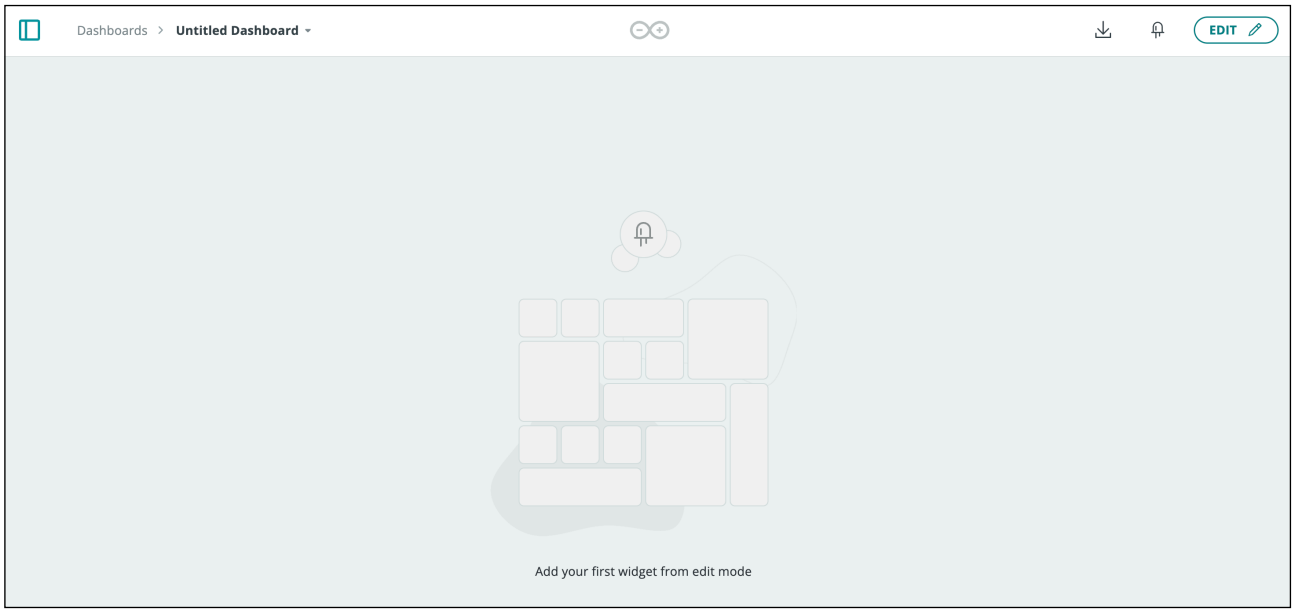
Figure 22: create dashboard



Naming the dashboard

We get a pretty blank screen. Let's give it a name. Click on `Untitled Dashboard` and give it the name `myLED`. Then click on `EDIT` (top right-hand corner).

Figure 23: the naming

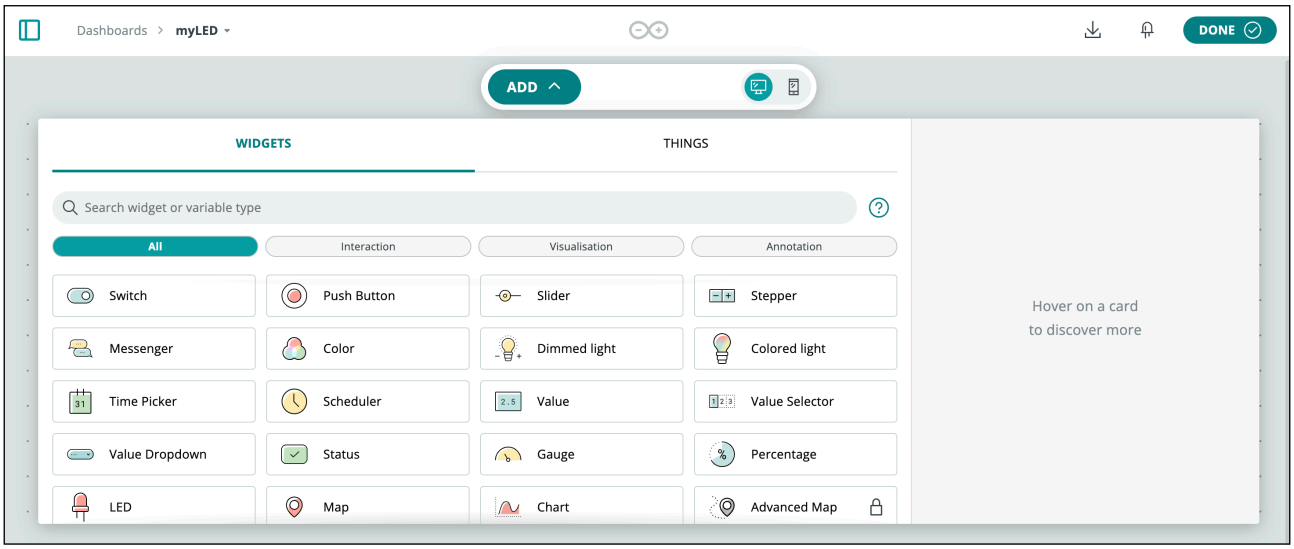


Editing the Dashboard

We have a blank dashboard, and this is where we add our widgets. There are a lot of widgets you can add for different purposes and components. For this, we only want one, a switch. Click on ADD, which should be highlighted after clicking EDIT. You can scroll to see them all. Some are only accessible if you have upgraded to a paid plan.

Click on the first one, Switch.

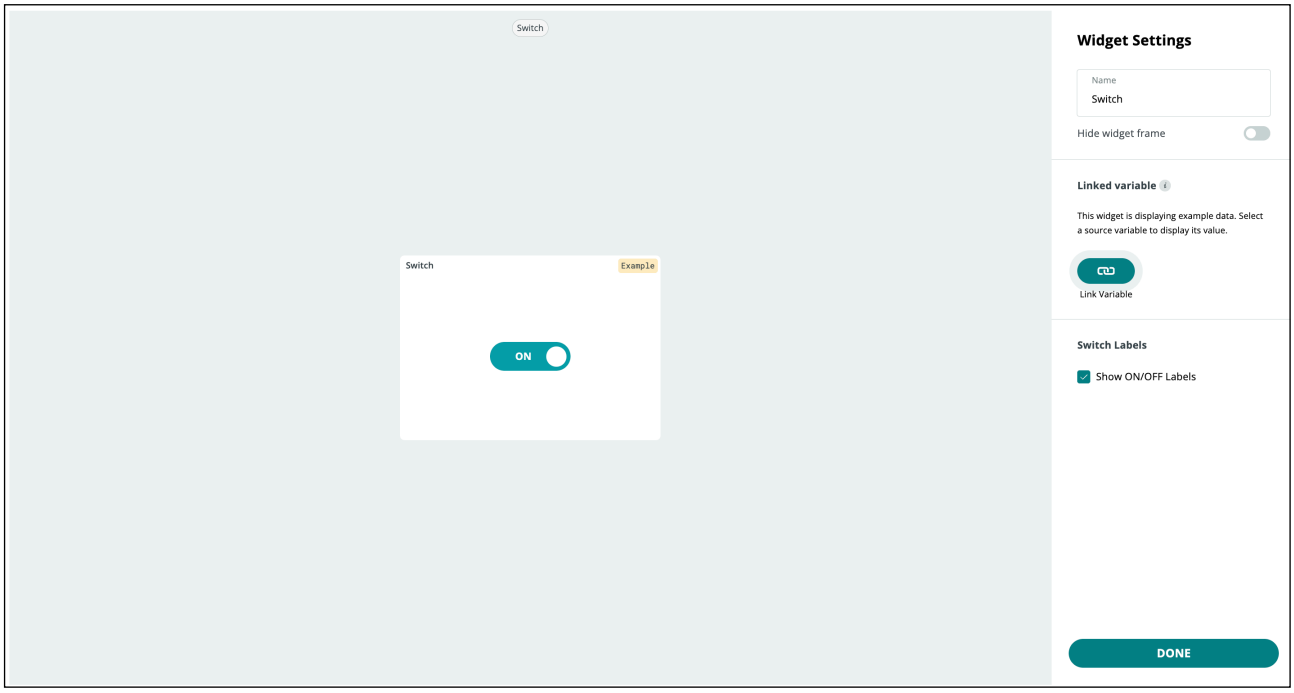
Figure 24: the list of widgets



The switch widget

You will get a page like the one below. This is where we can edit it a little bit, but more importantly, we connect it to the variable that we want to control. For now, leave everything else as it is and click on the icon for `Link Variable`.

Figure 25: widget setup



Linking the widget

You will be presented with a page shown below. It should have the name of myLED Thing and the name of the variable myLED. We could've been more creative with the naming, I know! Click on the myLED variable and then LINK VARIABLE to link it to the widget.

Figure 26a

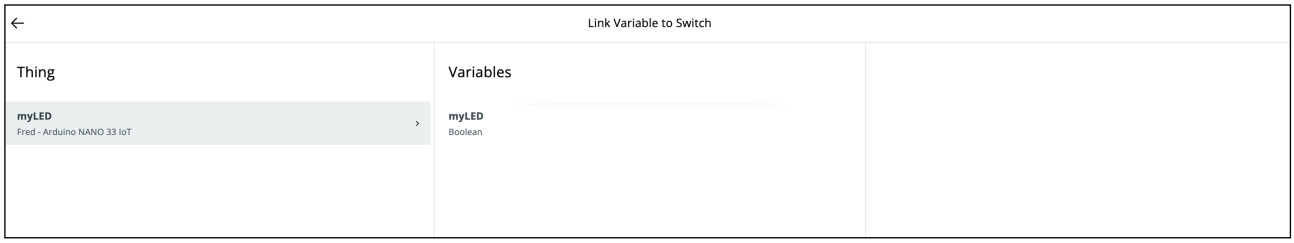
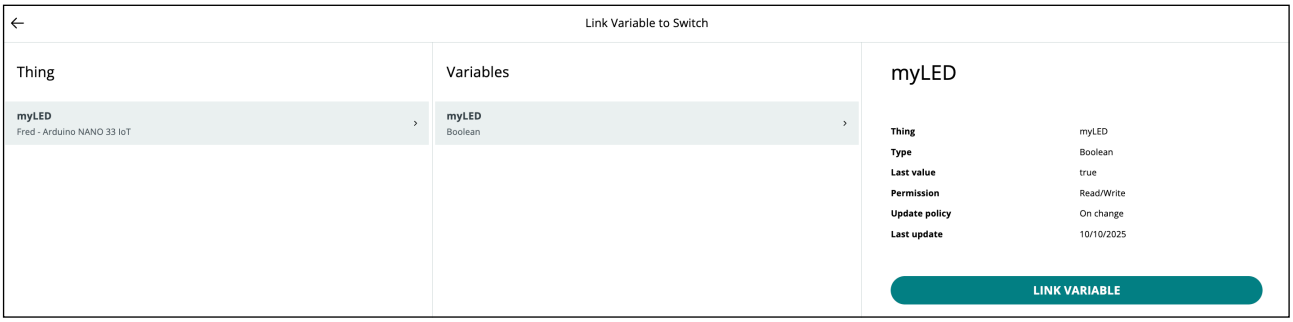


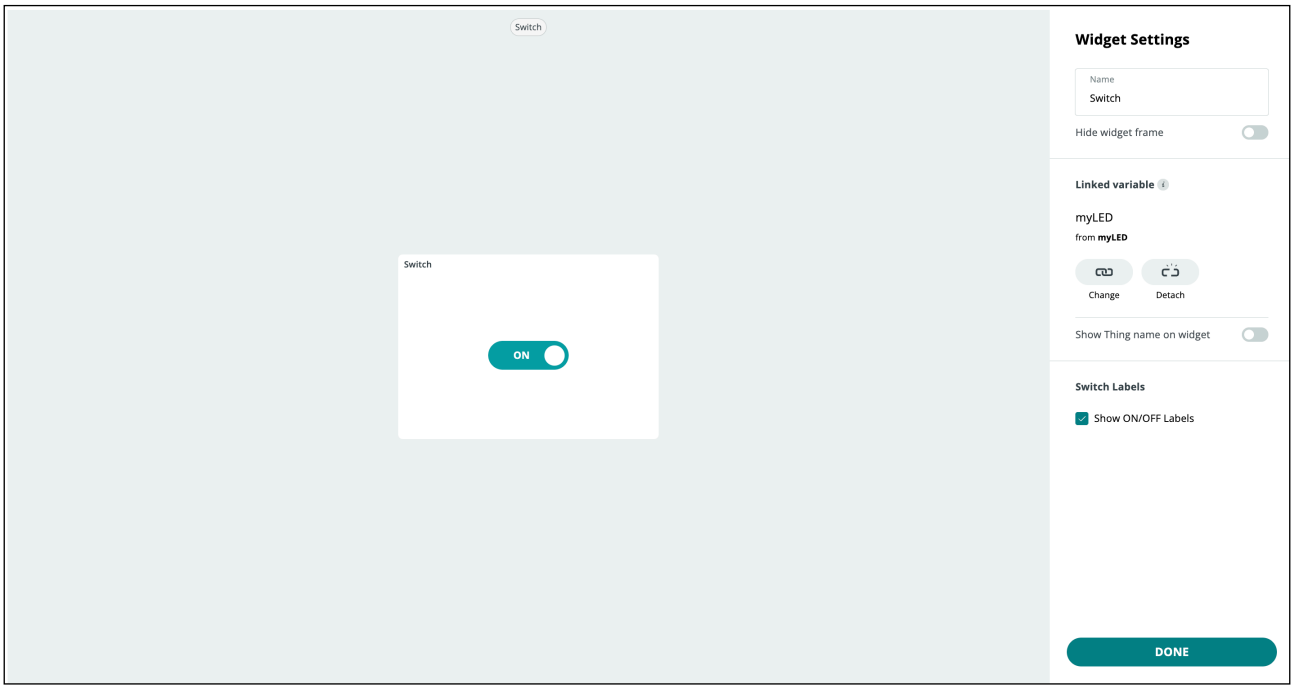
Figure 26b



Linked variable and widget

We are now back with the widget. If it has worked, then click DONE. You have do this before you can use it!

Figure 27



Testing the widget

First off, we should see our widget on the Dashboard. Click DONE, which means you aren't adding any more widgets. We can add more later. You can also edit the size and position, but for now, we just need to check it works.

Figure 28a

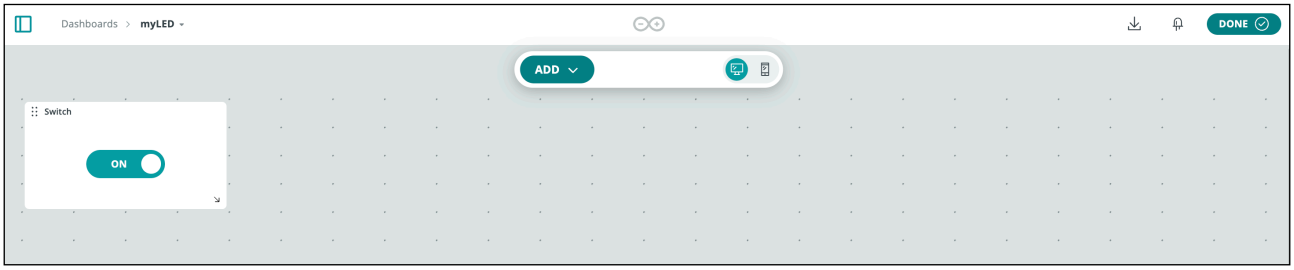
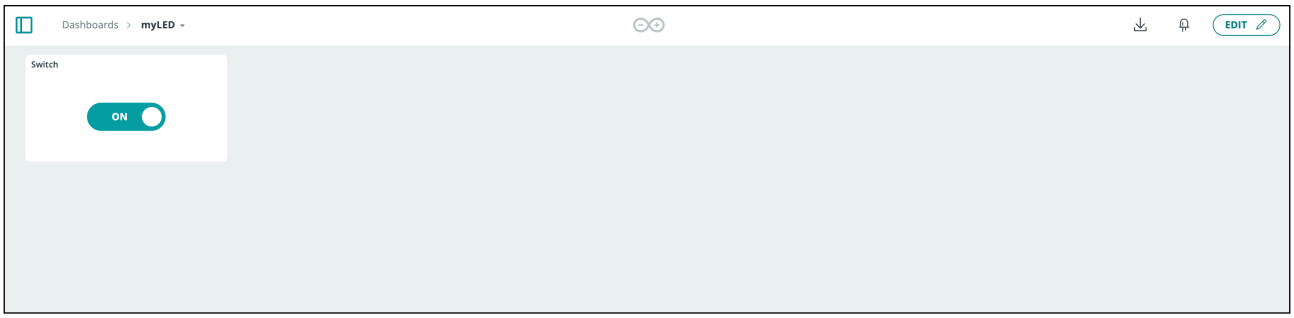
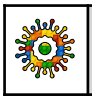


Figure 28b





Switching on/off

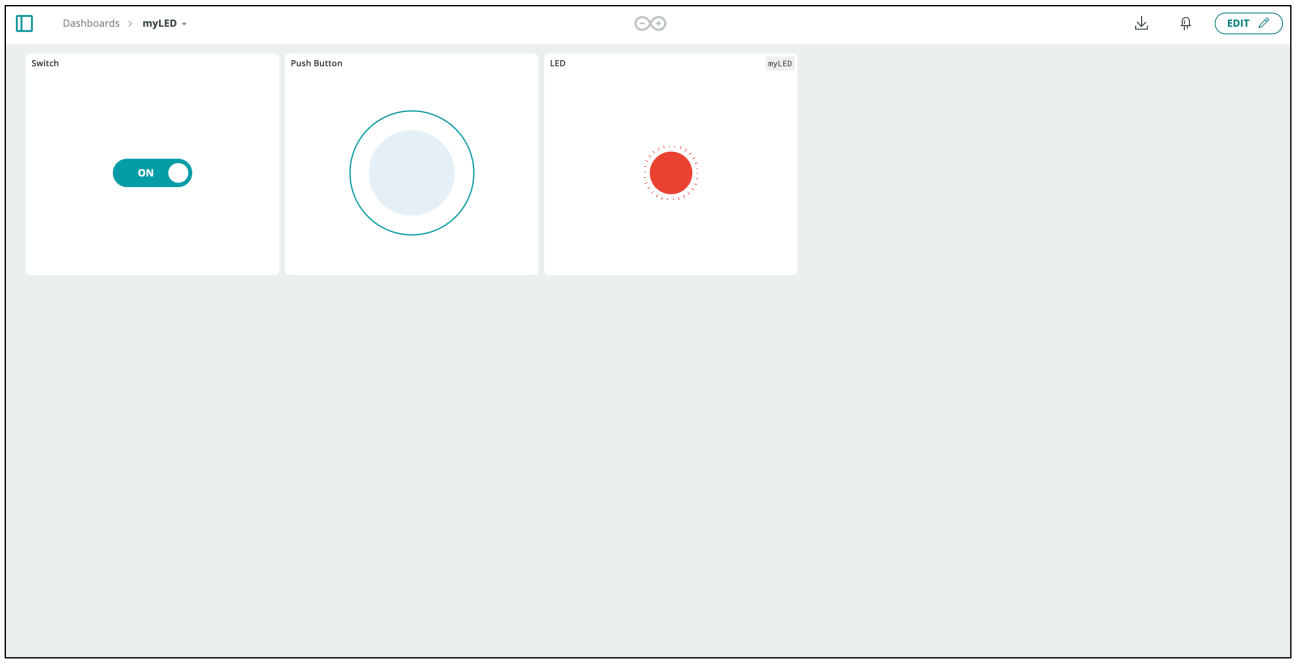
This should now work. Click on the widget to switch it on and watch the LED(s) switch on. Click again and it will switch it off. Assuming it worked for you, then you are controlling the device (the LED) over the WiFi from anywhere in the world. You don't even need to be on the same network.

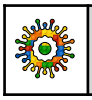
One way to test it is to plug your device into another power source. So it is not connected to your computer. Another way is to connect your computer to a different WiFi network, for instance, your phone, and test it. On both occasions, it should work.

Challenges

1. Add some other widgets, for instance, the push button and the LED (see below).
2. Change the size of the widget area.
3. Could you add another LED and use a different button?
4. Download the app for your phone and use it from there.

Figure 29: more widgets





Troubleshooting

I hope that you sailed through this, but I have found it a bit of a challenge, even creating this tutorial, mainly because I had already used it and tested it out first.

Issues:

- I found that it just wouldn't appear online.
- It was online, but the LEDs wouldn't respond to the switch despite the code and everything being correct.

Solutions:

If you encounter a major problem and you have double-checked, reloaded, and still it doesn't work, I found that if I uploaded a simple sketch using the download version of the IDE (not cloud, but cloud might work), for instance, the blink sketch, and then went back to the cloud and the above tutorial, it worked fine even though I did nothing different. I suspect that there was residual software, and so uploading a basic blink sketch just reset it.

Also, always double-check that the WiFi name and password are correct in secrets. It seems to default to something else for some reason. I hope that helps, but if not, don't give up.