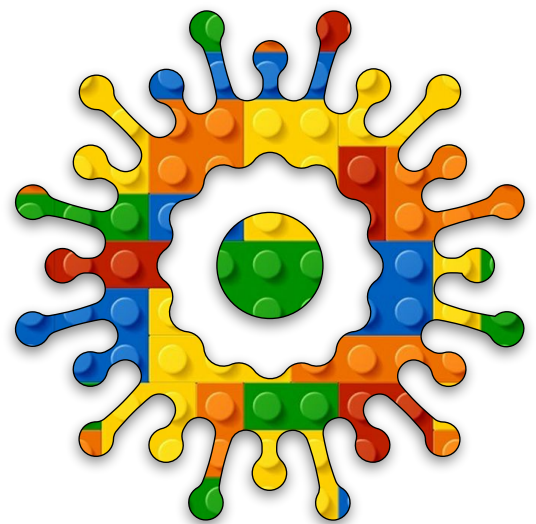


Internet of Things

Module B

Unit #2

fade LED





Module B Unit #2 fade LED

Simple steps

Sketch B2.1 the full sketch

Sketch B2.2 simplified sketch

Dash to the Dashboard

Widgets

Widget #1 Slider

Back to the Dashboard

Widget #2 Stepper

Widget #3 Value

Widget #4 Value Selector

Widget #5 Value Dropdown

All of them



Introduction to fade LED

Now that we have built our first Thing, we can build more. This means we can practice the somewhat convoluted process. This also means we don't have to go through the process of setting up the device again.

In this example, we are going to control the brightness of the LED (red on pin 2). For that, we need a variable; we will call that variable `val`, and it will receive a value between 0 and 255. This will be like turning the pot to make the LED brighter or dimmer.

We set up the LED as usual in setup.

```
pinMode(2, OUTPUT);
```

In the function that is `void onValChange()`, we put the following code in that will write to the LED the new value between 0 and 255.

```
analogWrite(2, myLED);
```



Simple steps

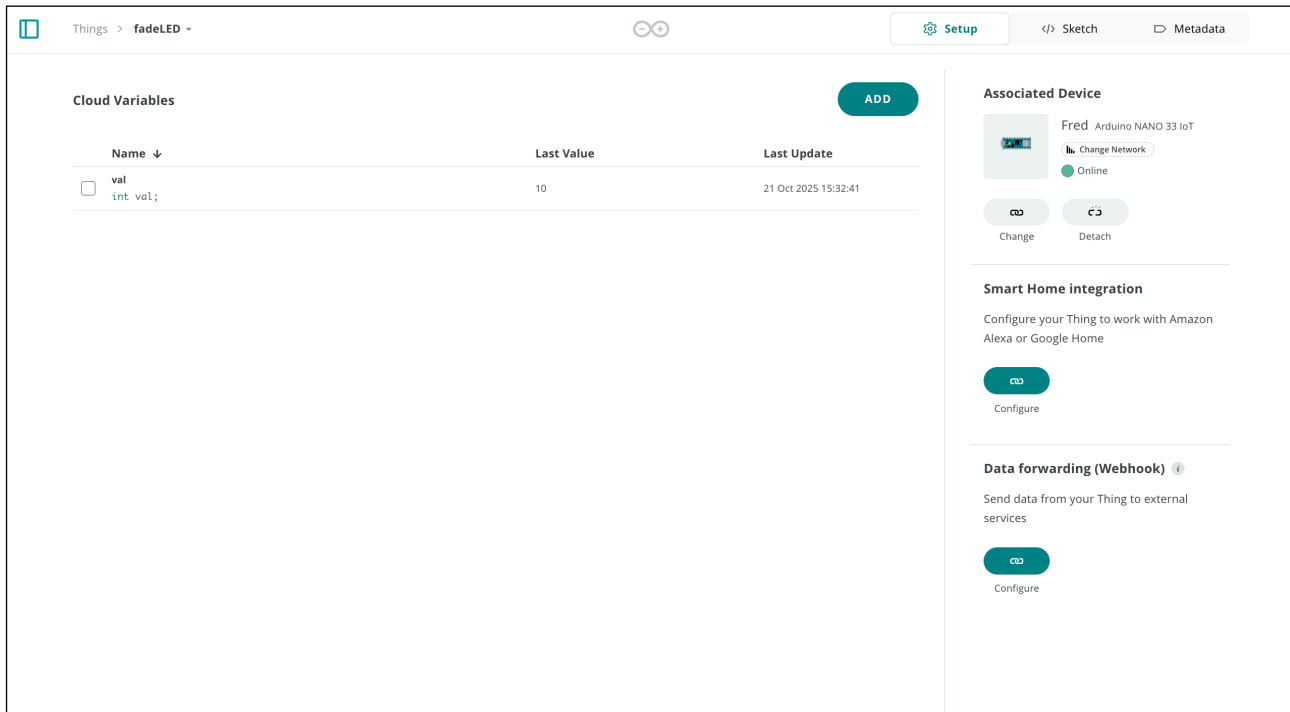
- 1 The first step is to delete the **myLED** Thing and Dashboard but keep the device as is.
- 2 Next, create a new Thing and call it **fadeLED**.
- 3 We need to add a variable. For this, we are going to add a basic variable that is an integer (**Integer Number eg. 1**). We will call the variable **val**. Then, **ADD VARIABLE** when ready.

Figure 1: adding the variable

The screenshot shows a dialog box titled "Add variable" with a close button (X) in the top right corner. Inside the dialog, there is a text input field labeled "Name" containing the text "val". Below this is a button labeled "Sync with other Things" with a circular arrow icon. Underneath is a section titled "Select variable type" with a dropdown arrow. This section contains several category buttons: "Basic types" (which is selected with a checkmark), "Energy", "Light and color", "Size and motion", "Smart home", and "Time". Below these buttons is a list of variable types: "Boolean eg. true", "Character String eg. 'Hello'", "Floating Point Number eg. 1.55", and "Integer Number eg. 1". The "Integer Number" option is currently selected. At the bottom of the dialog, there is a section titled "Variable Update Policy" with two radio buttons: "On change" (which is selected) and "Periodically". At the very bottom of the dialog are two buttons: "CANCEL" and "ADD VARIABLE".

4 Then we attach the device we already have.

Figure 2: attaching the device



5 Next, we go to the `</>` **Sketch** tab at the top.

6 You should have the basics of the sketch. We want to add the relevant parts that will make it work. The great thing is there are only two lines of code to add!

7 Just make sure that you have the Secrets tab, if not then click the + **Add Tab** button, and that the right information is in there for the **WiFi**.

8 When you have added the code (see **Sketch B2.1** below), click on the upload button.



Sketch B2.1 the full sketch

I have highlighted the changes, and the code has been added. Upload when you have added the code.

```
/*  
  Sketch generated by the Arduino IoT Cloud Thing "Untitled"  
  https://create.arduino.cc/cloud/things/xxxx  
  
  Arduino IoT Cloud Variables description  
  
  The following variables are automatically generated and updated  
  when changes are made to the Thing  
  
  int val;  
  
  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);  
  pinMode(2, OUTPUT);  
  // 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 Val is READ_WRITE variable, onValChange() is
    executed every time a new value is received from IoT Cloud.
*/
void onValChange() {
    analogWrite(2, myLED);
}

```



Notes

It can be a bit messy with all the blank lines and comments. Below is a cleaned-up version.



Sketch B2.2 simplified sketch

Taking out all the commented bits and rearranging the furniture, as you might say.

```
#include "thingProperties.h"

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

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

void onValChange()
{
  analogWrite(2, val);
}
```



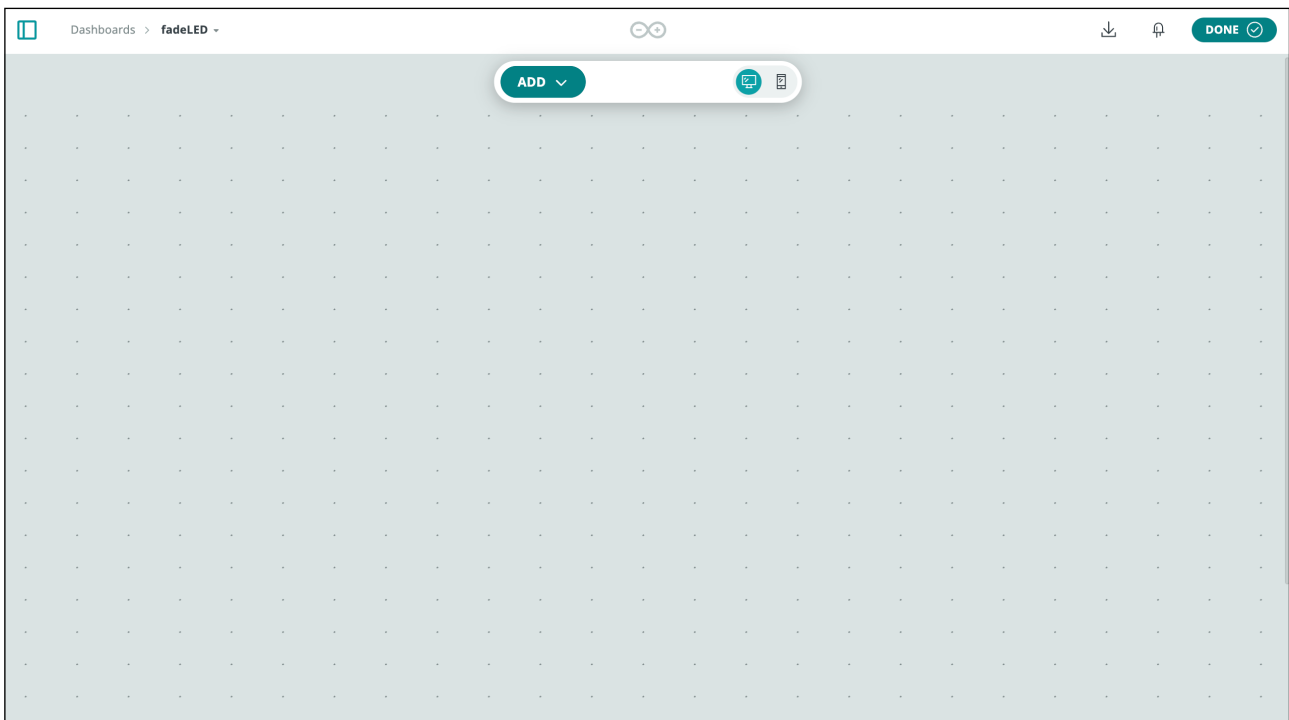
Notes

Seems much simpler, cleaner, clearer.

Dash to the Dashboard

We will now add a widget to control the LED. Head over to Dashboards and change the name from **Untitled** to **fadeLED**.

Figure 3: naming





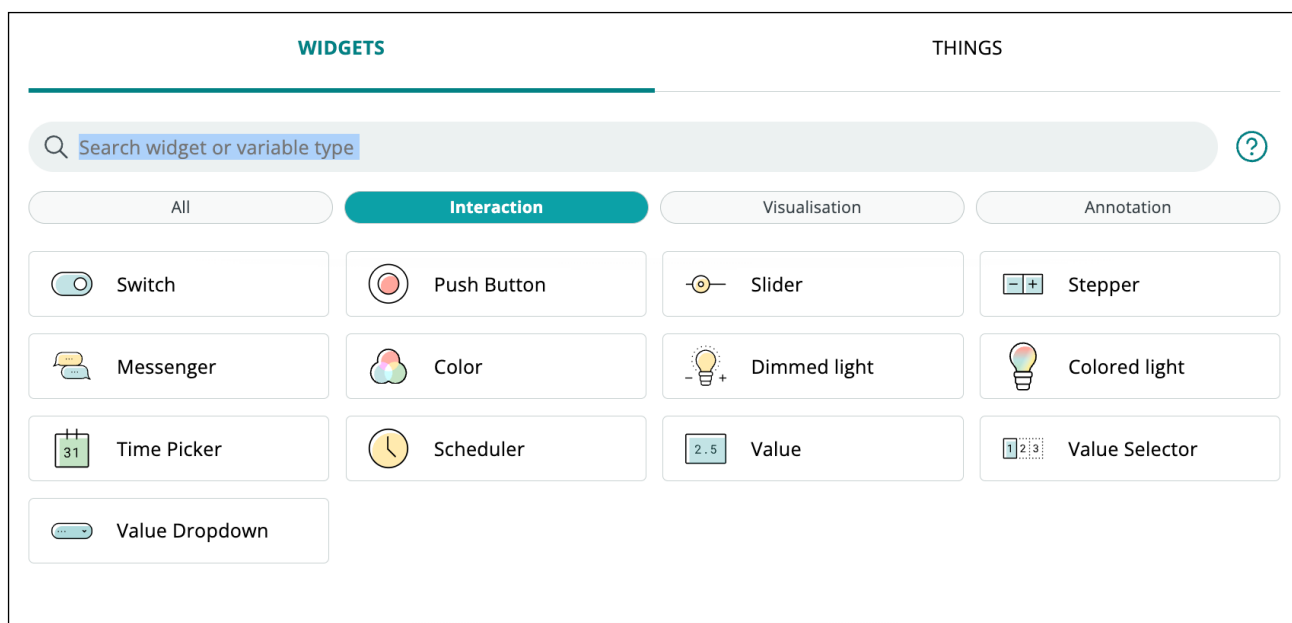
Widgets

Then we **ADD** a widget. I've selected the interaction tab so we can see all the widgets associated with interacting with components. There are five that we can use.

- 1 Slider
- 2 Stepper
- 3 Value
- 4 Value Selector
- 5 Value Dropdown

The last two are very similar.

Figure 4: widgets



Widget #1 Slider

Select the **Slider** widget.

Figure 5: Slider widget

Slider

Slider

Example

25

0

50

Widget Settings

Name

Slider

Hide widget frame

Linked variable ⓘ

This widget is displaying example data.
Select a source variable to display its value.

Link Variable

Orientation

☒ Horizontal

☐ Vertical

Value range

Min

0

Max

50

DONE

We need to link the variable to **val** (click on the chain symbol) and click on **val** when it appears next to **fadeLED** Thing.

Figure 6: linking the variable

←

Link Variable to Slider

Thing

fadeLED

Fred - Arduino NANO 33 IoT

Variables

val

Int

val

Thing

fadeLED

Type

Int

Last value

10

Permission

Read/Write

Update policy

On change

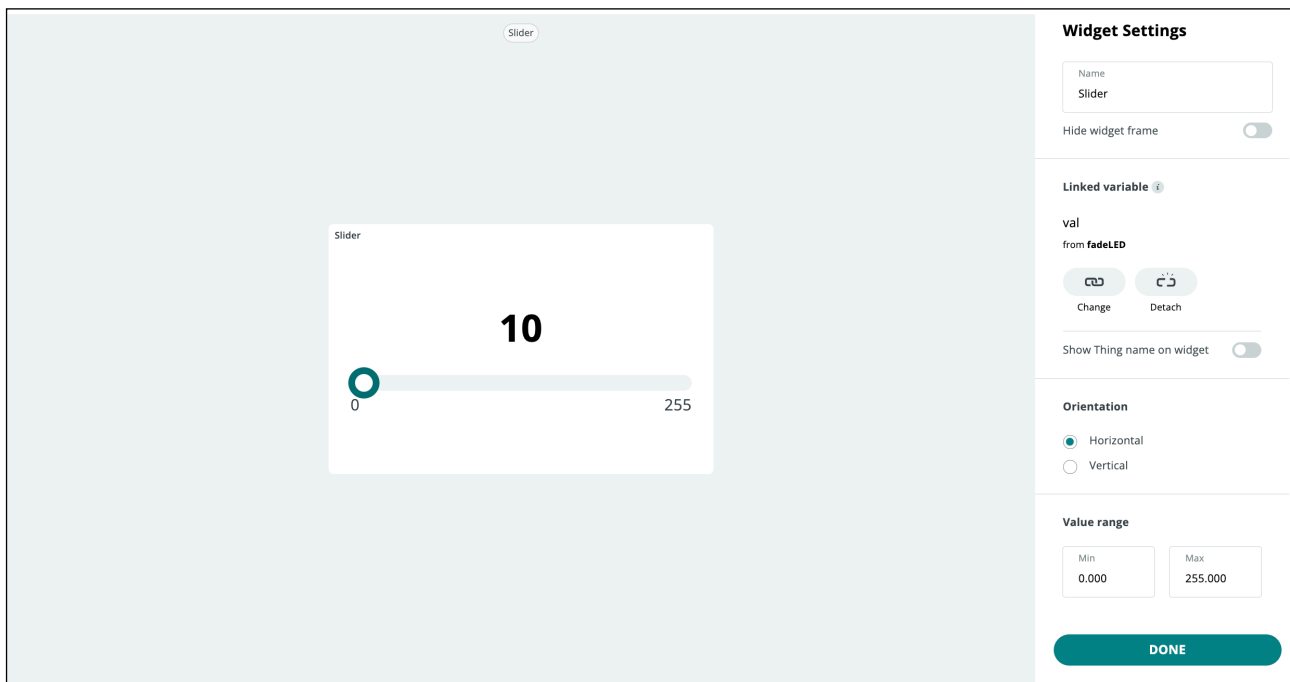
Last update

10/21/2025

LINK VARIABLE

Once the variable is linked, we are taken back to the widget setup. We change the **Value Range** from **0.00** and **50.000** to **0** and **255**, then click **DONE**.

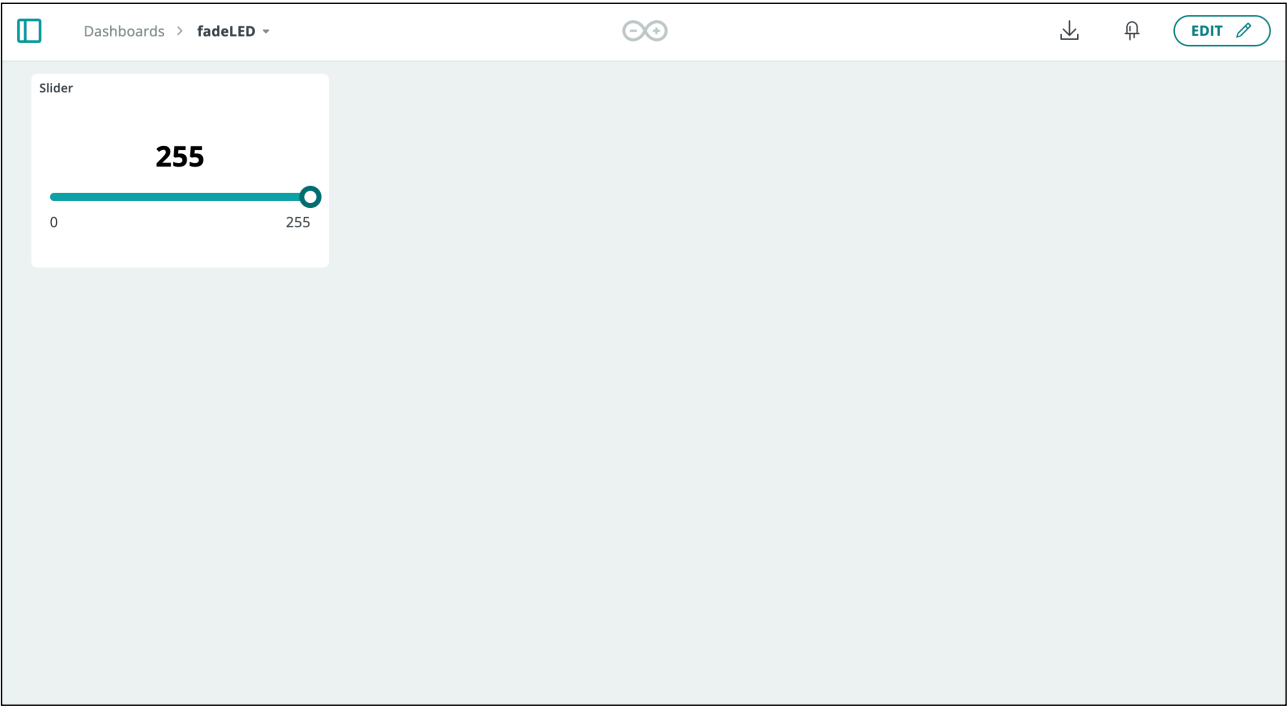
Figure 7: widget setup values



Back to the Dashboard

Back with the main dashboard. Click on **Done**, and when you slide the slider, it should change the brightness of the LED accordingly.

Figure 8: slider



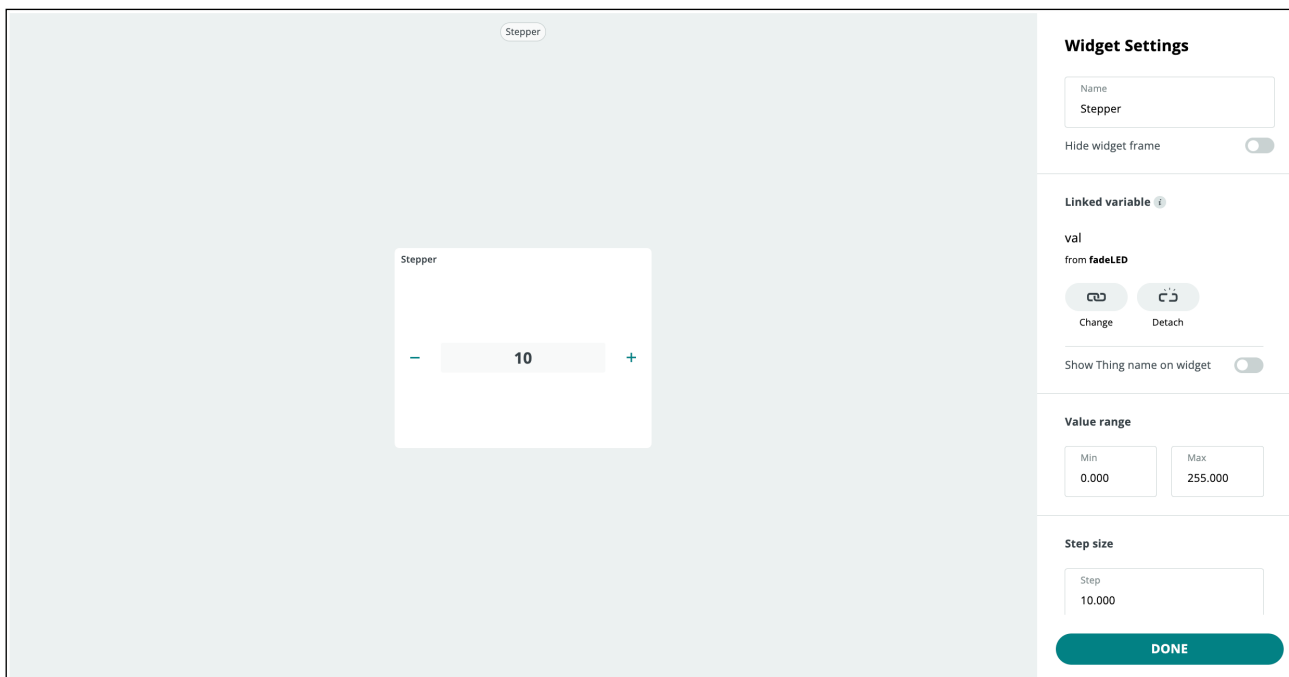


Widget #2 Stepper

Click Edit, select the slider widget and click delete (the red bin). Let us add a new widget, the **Stepper**. You will need to link it to the **val** variable as before. Change the **min** and **max** value range to **0** and **255**. Also, the **Step Size** to **10** (a step size of **1** is quite tiresome). Click **DONE** and **DONE**.

! You can click too fast, in which case it will miss some of the steps out. Increase the **step size** to **25** (or more) for more rapid change per step.

Figure 9: stepper



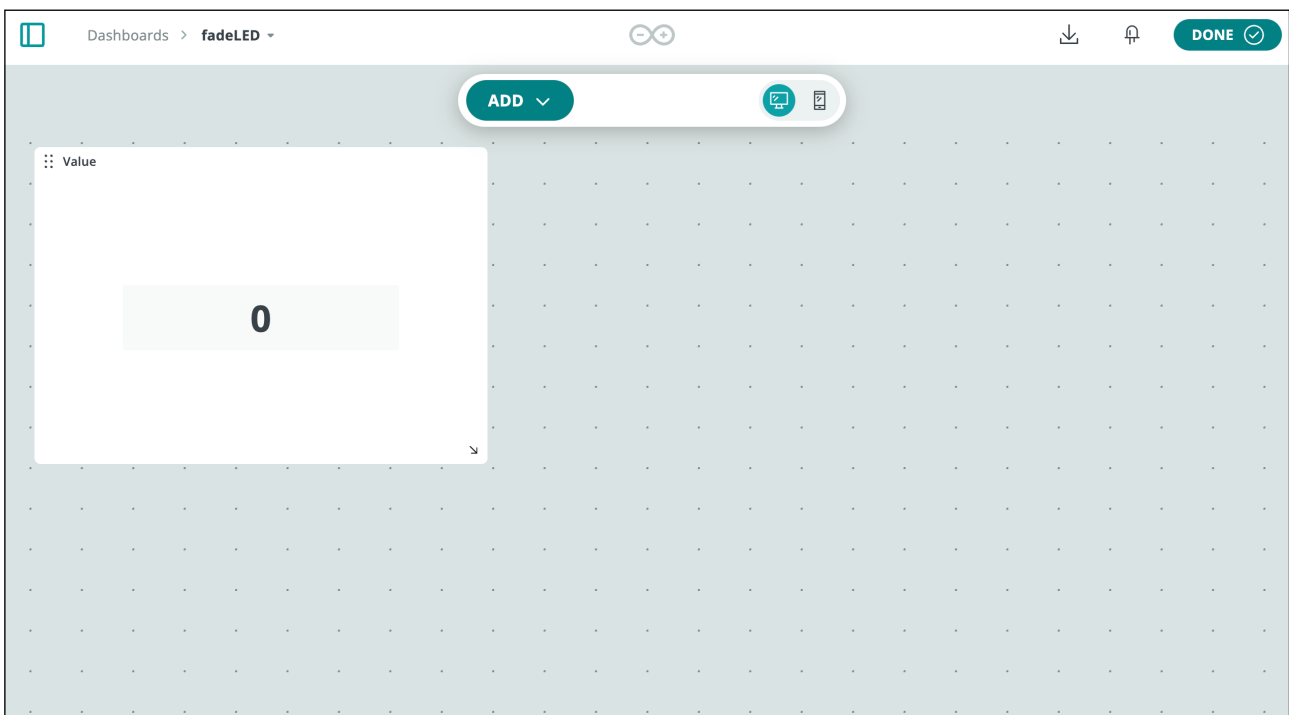


Widget #3 Value

Delete the **Stepper** widget (click **Edit** and then red bin). Add the **Value** widget from the list. This means you will have to enter a value for the LED. Just link the variable and click **DONE**. There are no other options because it is an integer. When you click on the value inside the widget, change it and press return. This will change the brightness of the LED accordingly.

! While in Edit mode, you can grab the bottom right-hand corner and stretch the shape and size of the widget. You can do this with most widgets.

Figure 10: size and shape of Value widget

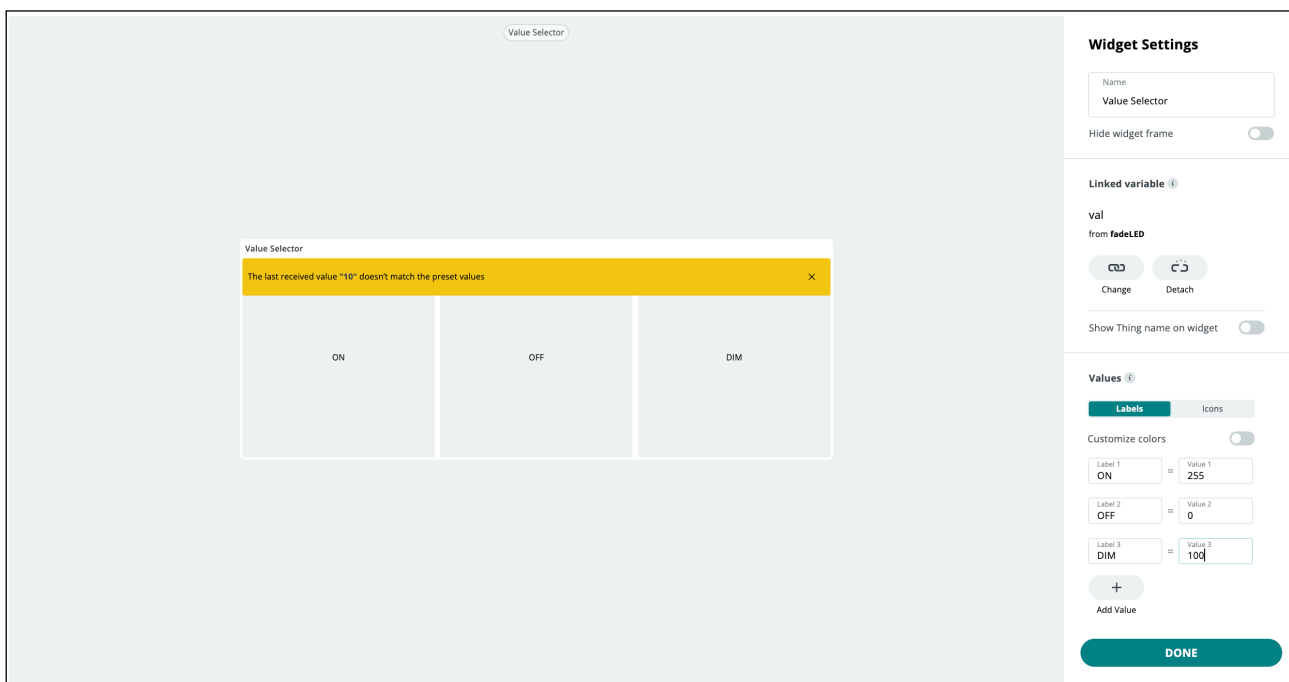




Widget #4 Value Selector

We can assign values to what look like switch/buttons you can click on. You can add more buttons, but we will use the three (minimum). Delete the **Value** widget. Select the **Value Selector**. You will be given three selectors. We will change the names and the values. For **ON**, we will give it the value of **255**, **OFF** is **0** (no change), and for **AUTO**, we will change the name to **DIM** and give it a value of **100**. You will get a temporary error message. Click **DONE**. When you select the appropriate button/switch/selector, this will switch the LED brightness accordingly.

Figure 11: Value Selector

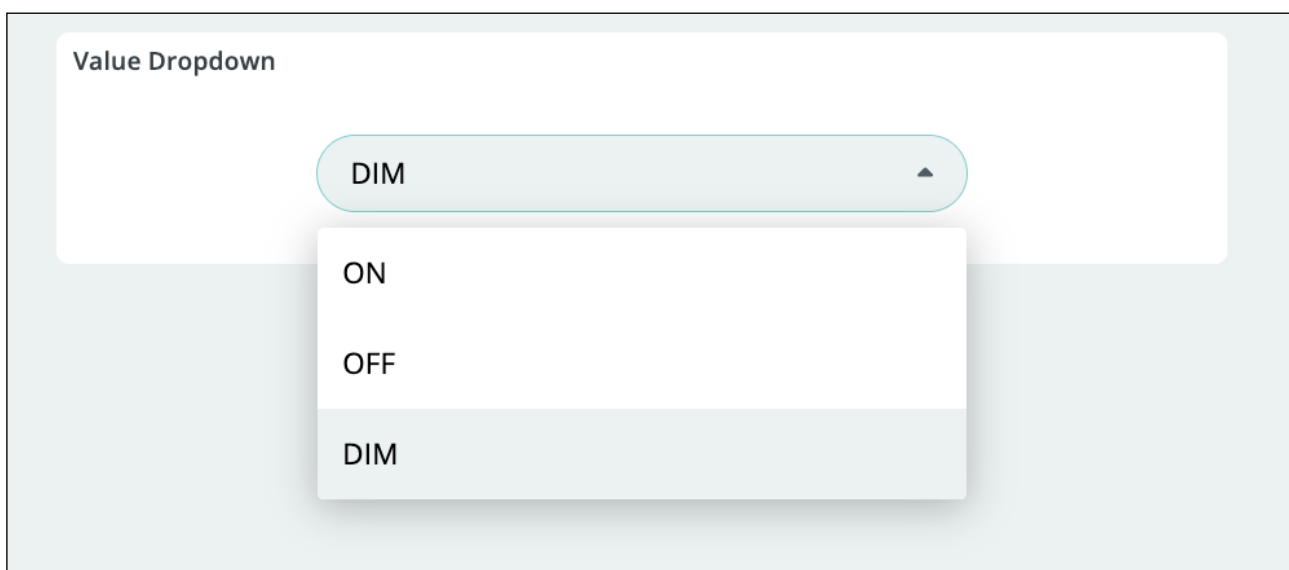




Widget #5 Value Dropdown

Delete the previous widget and select **Value Dropdown**. This is very similar to the previous widget. Try different names and values. Click **DONE** and **DONE** when done. You click on the small arrow to get the drop-down menu and select what you want, which will in turn change the LED accordingly.

Figure 12: Value Dropdown





All of them

Just to finish off, I will show you all the widgets on one dashboard. You can arrange them however you want, link all of them to the variable. Play around with them and change them.

Figure 13: all the widgets

