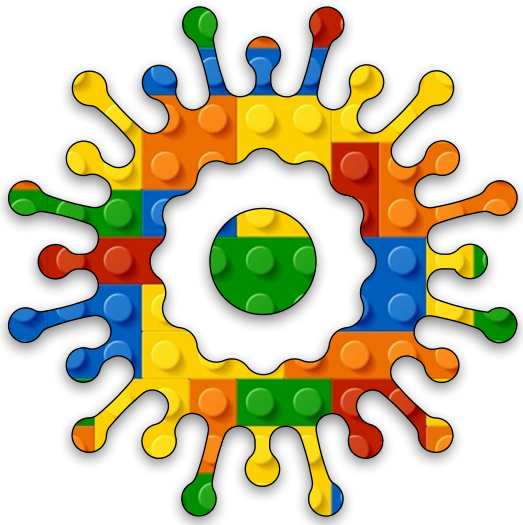
Internet of Things Module B Unit #4 the potty



Content

Module B Unit #4 the potty

Introduction to potty

For this unit, we will use the pot, the variable resistor we used in the module A unit #9 the pot. The circuit diagram and setup are shown below. We use the same analog pin A0 as before.

Figure 1a: pot circuit diagram

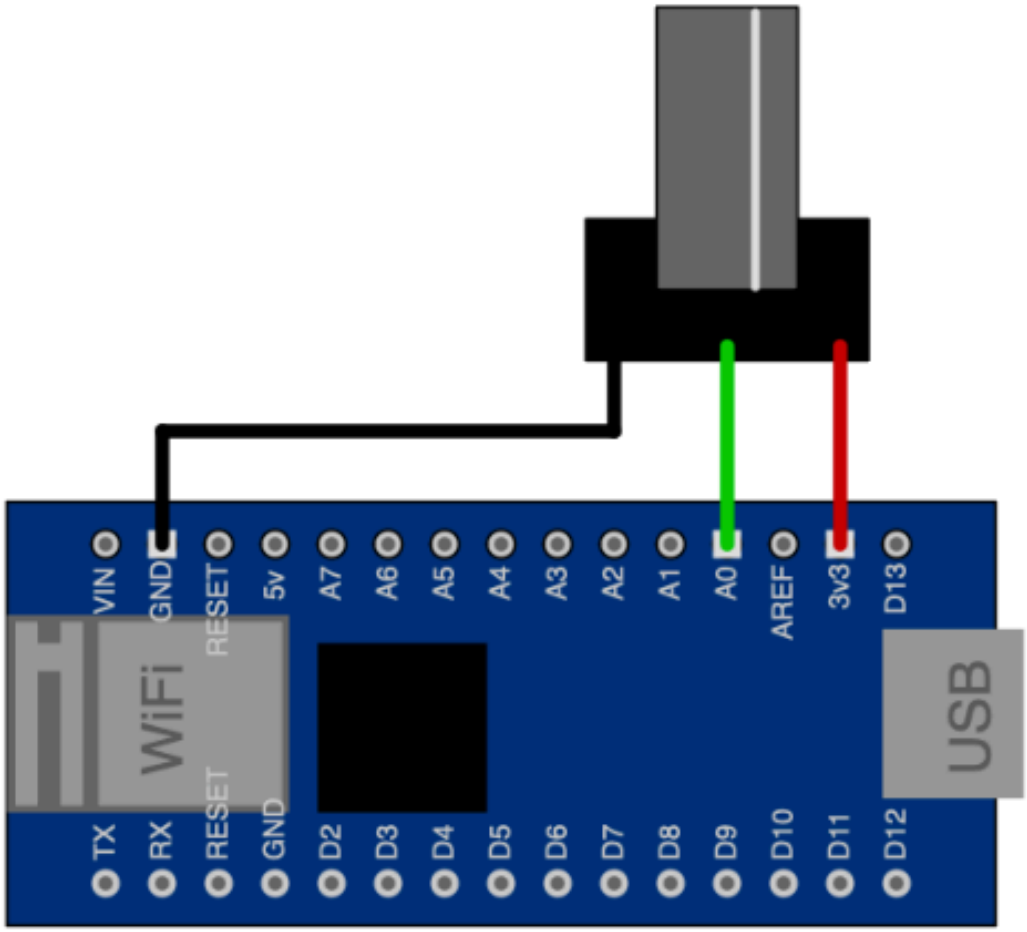
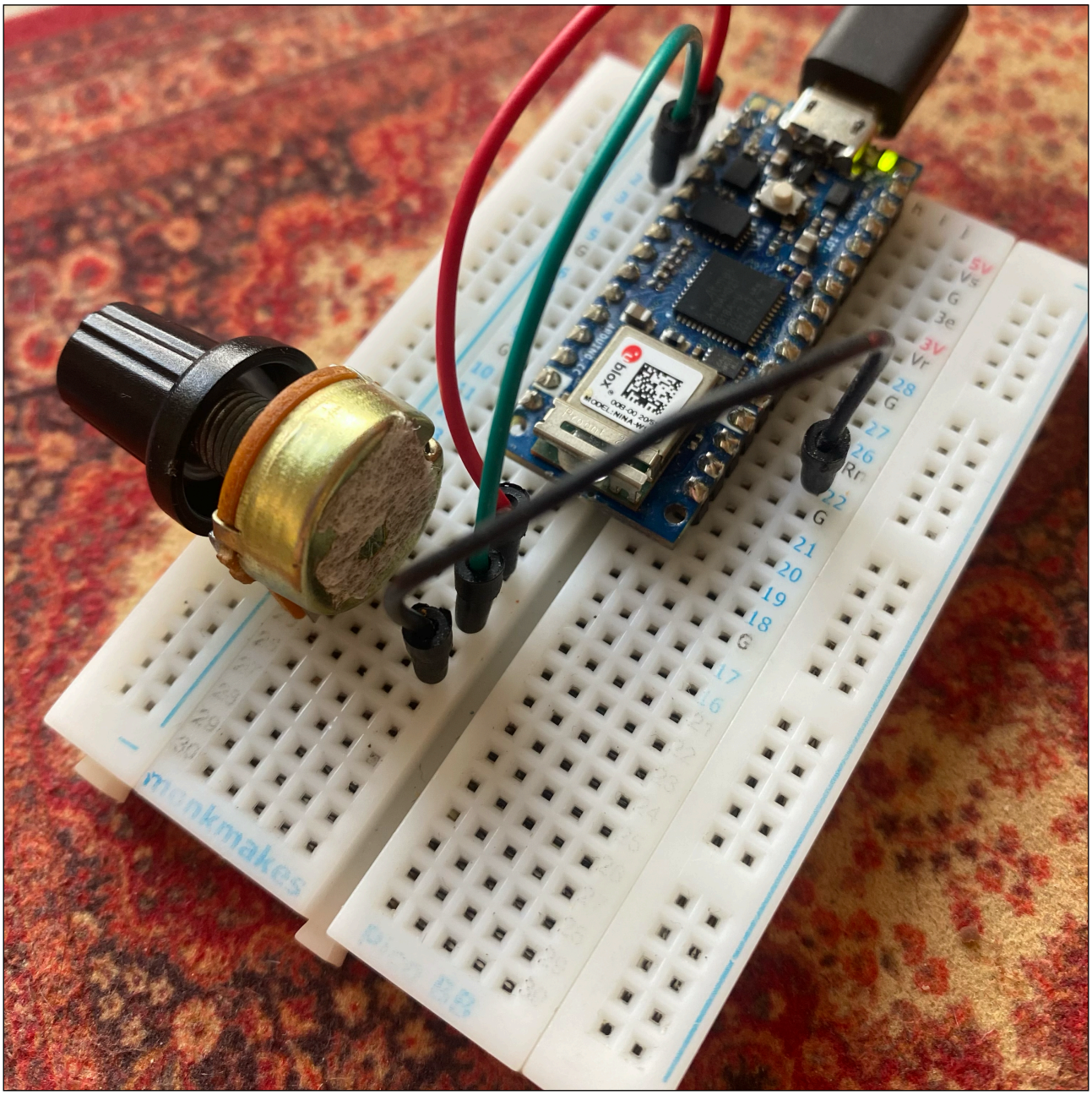


Figure 1b: set up with the pot



The Thing

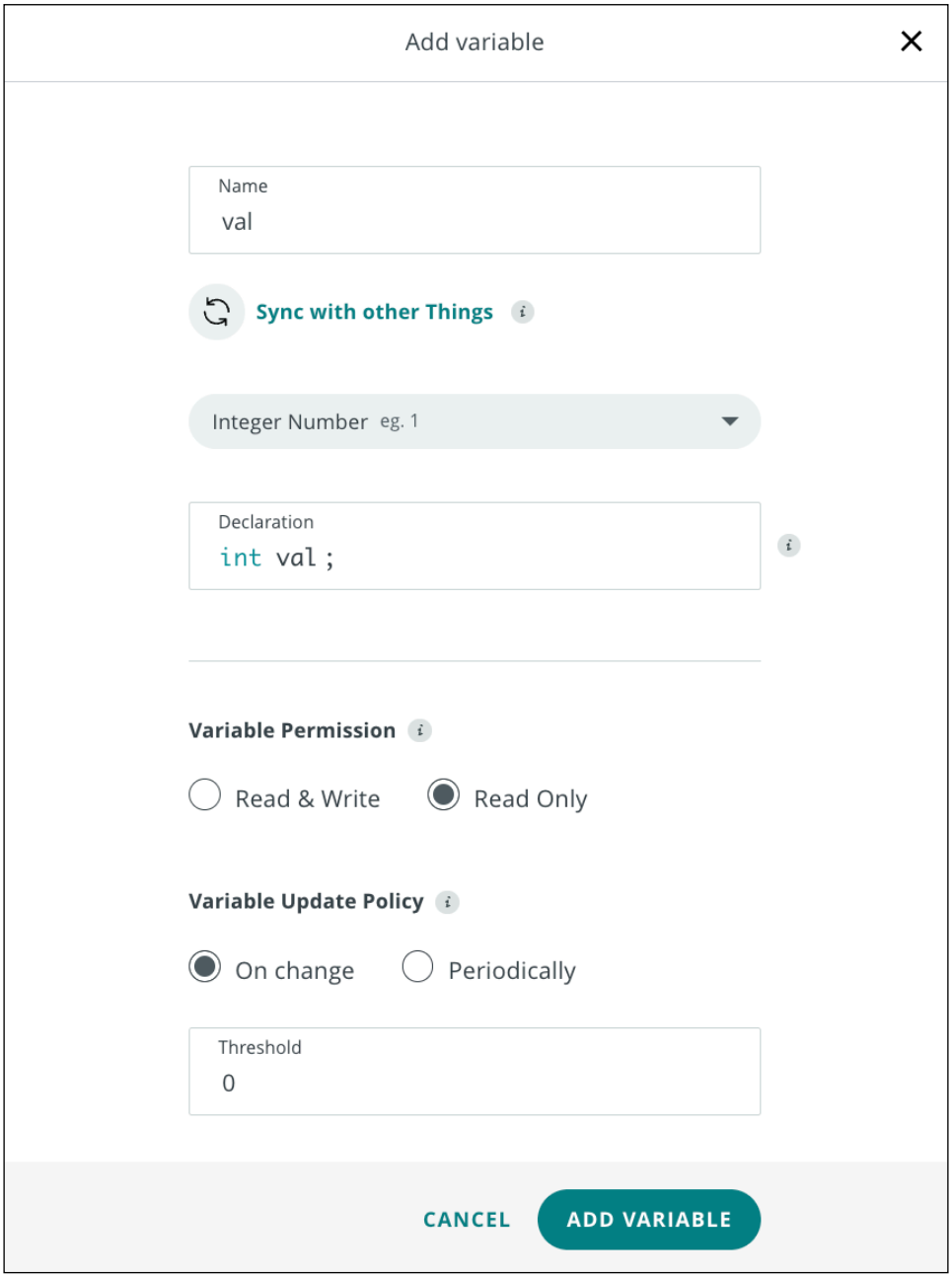
We delete the previous Thing and Dashboard.

1. Create a new Thing and call it myPot.
2. Attach your device.
3. Add the variable...

The variable

We are going to add a variable called `val`. The variable will be an integer and it will be `Read Only`.

Figure 2: adding the variable

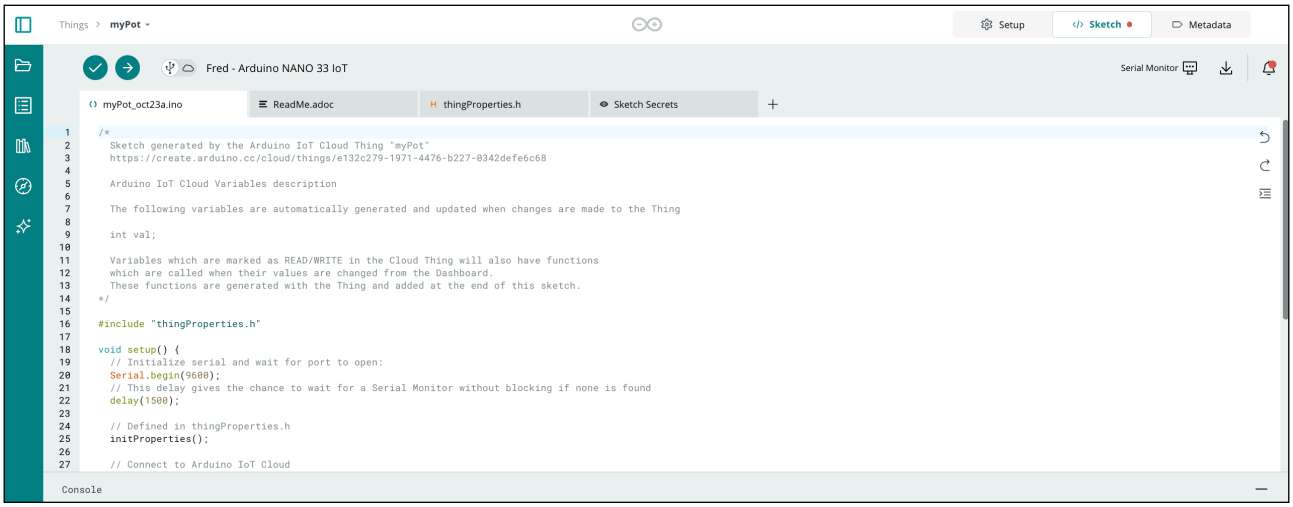


The sketch

We have our Thing and our variable `int val`. Now click on the `</> Sketch` tab to get our sketch.

❗ If, like me, you don't have your `Sketch Secrets` tab, then go to `Devices` and check if it is there. Then return to the sketch (which might still have the tab missing) and click `+` to add the tab (if still missing). Go into the tab and check if the credentials are OK.

Figure 3: the sketch



Sketch B4.1 the generated sketch

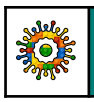
You should be getting familiar with this now.

```
/*
  Sketch generated by the Arduino IoT Cloud Thing "myPot"
  https://create.arduino.cc/cloud/things/xxxxx

  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);
  // 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


}
```

Adding the code

We don't need to specify that analog pin 0 is an input. Also, we want to read the value from that pin. We do this with the analogRead() function.

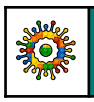


Sketch B4.2 full sketch

Adding in the code to read the analog pin (A0) using the variable val. Upload when added (remove commented and spaces).

```
/*
  Sketch generated by the Arduino IoT Cloud Thing "myPot"
  https://create.arduino.cc/cloud/things/xxxxxxxxxx

  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);
  // 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();
  val = analogRead(A0);


}
```



Sketch B4.3 simplified sketch

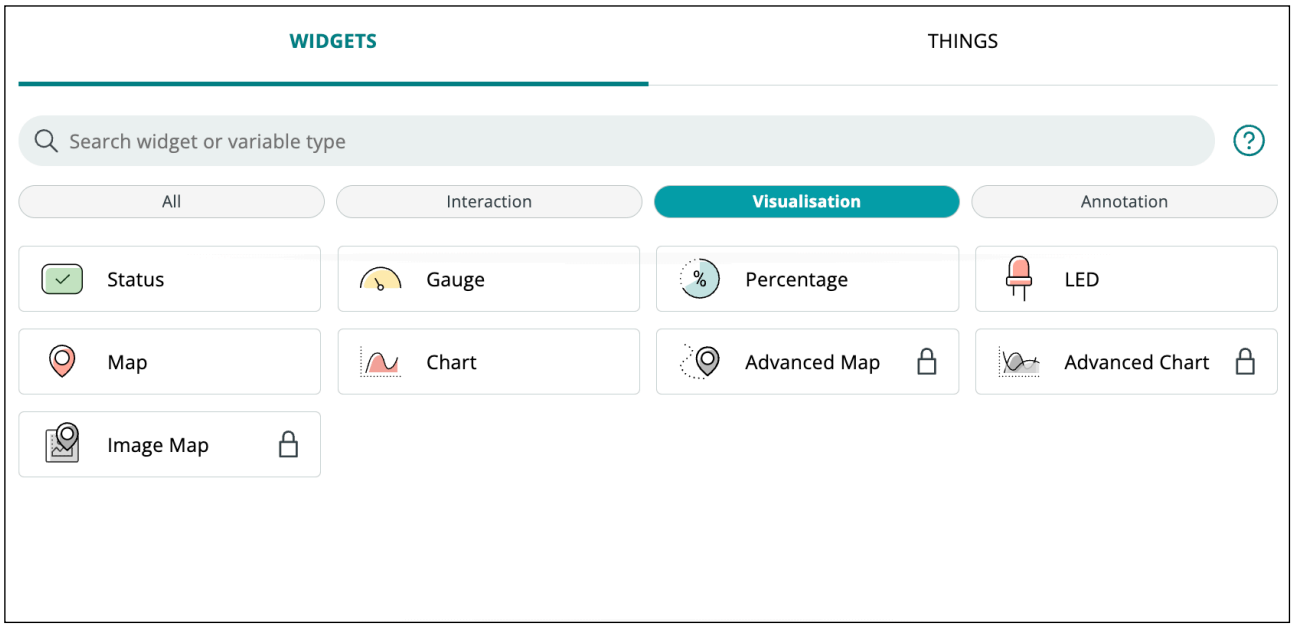
Removing the unnecessary comments. Upload.

```
#include "thingProperties.h"

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

void loop() {
  ArduinoCloud.update();
  val = analogRead(A0);
}
```

1 Create a new Dashboard (delete the old one if you haven't already) and call it `myPot`.

2 Add a widget, select `Visualisation`.

Figure 4: adding a visualisation widget

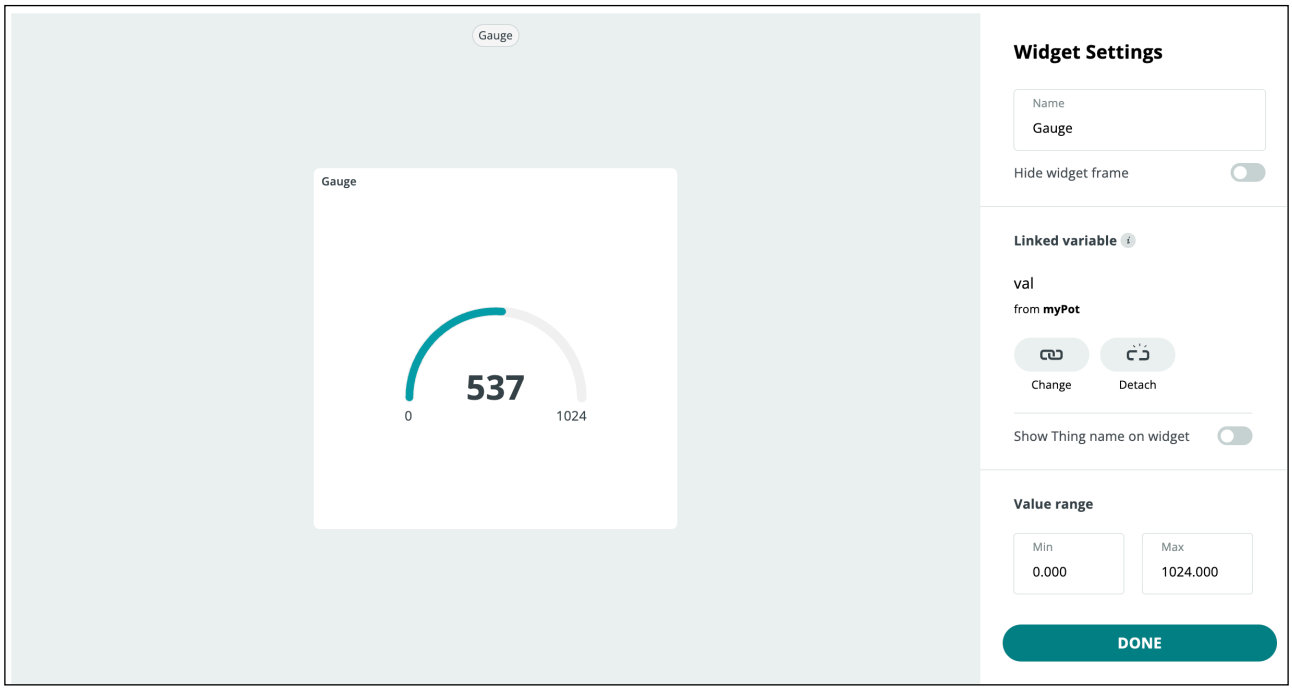


3️⃣ Select the Gauge widget.

4️⃣ Link to variable `int val`.

5️⃣ Make max range `1024` and click `DONE`.

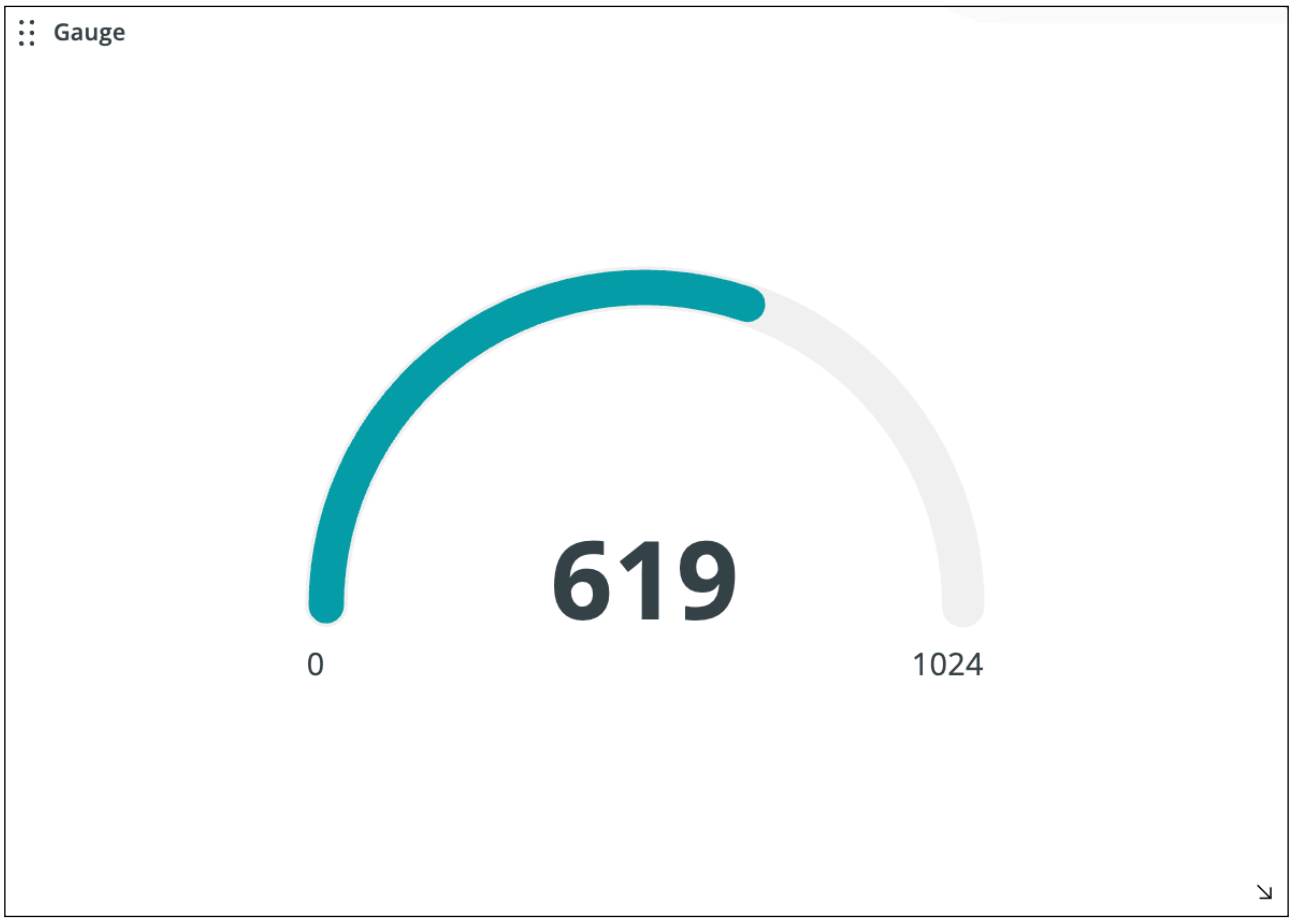
Figure 5: the gauge widget



Turn the pot

As you turn the pot, the gauge moves and the data is displayed. There is a slight lag.

Figure 6: turn the pot, see the gauge respond



Challenge

Have a go at the `Percentage` widget.