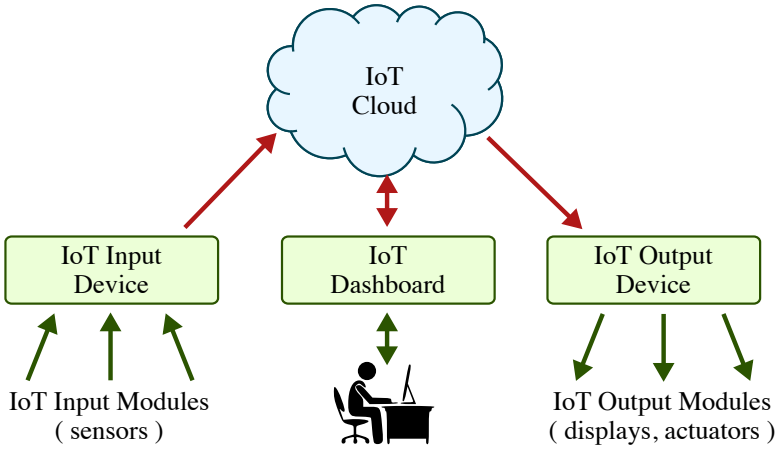
CURIE Academy, Summer 2021
Design Project Dashboard Tutorial

Nick Cebry and Christopher Batten
School of Electrical and Computer Engineering
Cornell University

Each IoT design project will involve building an IoT system comprised of an IoT input device, IoT cloud, IoT output device, and IoT dashboard. The IoT input devices will have various input modules attached that can sense what is going on in the environment (e.g., light, temperature, humidity, button press, rotation, water, acceleration, distance, soil moisture) and be able to upload data into the cloud. Each IoT output device will be able to download data from the could and will have various output modules attached to display data (e.g., LEDs, piezo buzzer, 4-digit display). The IoT dashboard can be used to monitor or even configure the IoT system. The following diagram illustrates the overall approach we will be using in our IoT systems.



The rest of this document describes how to setup the IoT dashboard. Revisit the Lab 2 notes and handout to learn about the Particle Argon ports and how to use the Particle development environment. See the Design Project Input/Output Tutorial for more on how to use the various input and output modules.

1. Sending Data to Ubidots

We will be using Ubidots to create our IoT dashboards. Ubidots provides data visualization and control for IoT products. Your IoT devices are designed to communicate with the Particle cloud, so we need a way to get the data from the Particle cloud to the Ubidots cloud. The course staff has already set up the connection between the Particle and Ubidots system, so all you need to do is send the data to the Particle cloud in a special way so that Particle knows it needs to pass it on to Ubidots. The example code below demonstrates how to send two values (a digital value from the button input and an analog value from the rotation sensor) to the Ubidots cloud. We will use a library to create the special events that Particle will know to pass on to Ubidots. The library can be added to your project by clicking the library button in the Particle IDE, entering "ubidots" in the search bar, and then clicking on the Ubidots library (NOTE: this is different from the UbidotsMQTT library) to add it to your project. Before the setup section, we add two lines of configuration for Ubidots. These will be exactly the same in every project you want to add a dashboard to. In this example code, we are sending a message to Ubidots once every second in the loop section. Each time we want to send a message, we add any variables we want to send, then call the send function.

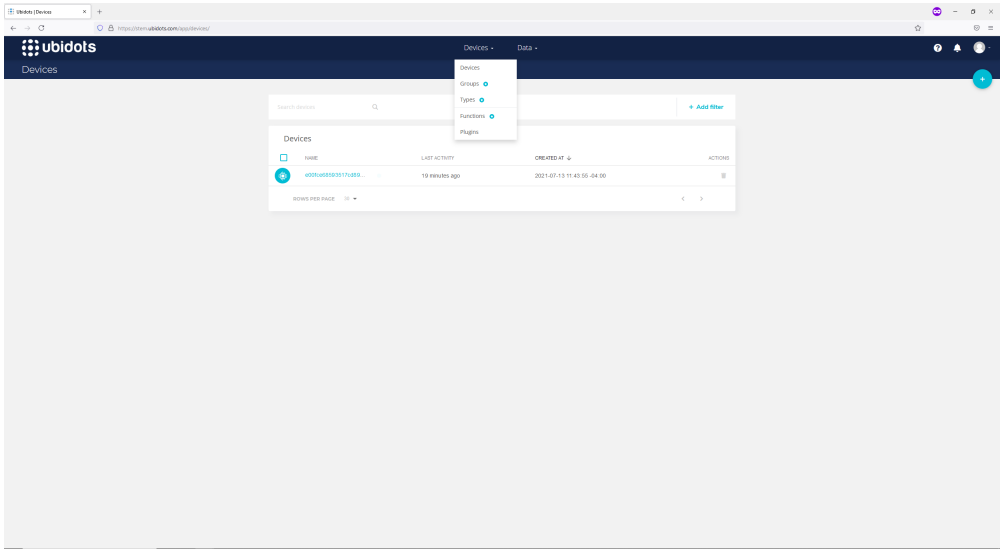
Program your device with the code below to begin sending data to Ubidots. If you look at the console, you should see events with the name "ubidots" and a data field with your variable names and their values inside of curly braces. This is your Particle Argon sending messages to the Particle Cloud that should be forwarded to Ubidots. You should also see events titled "hook-sent/ubidots", this is the Particle cloud letting you know that it has sent a message to Ubidots. Finally, you should see events called "hook-response/ubidots/0", which is Ubidots sending a message back to the Particle cloud letting it know that it got the data.

At this point, you may want to turn your Particle Argon off, or reprogram it with some other code, since there's a limit to how much data you can send to Ubidots in one day.

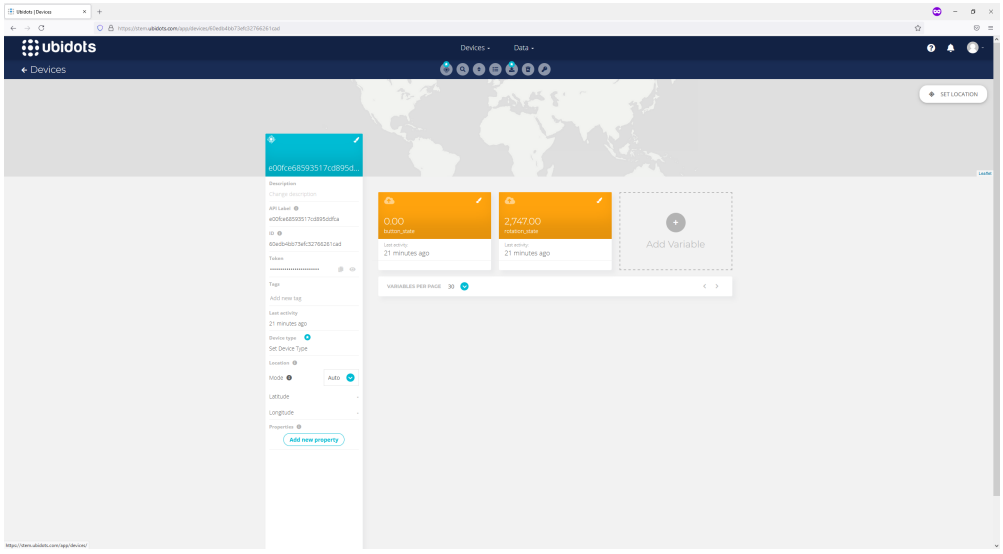
```
#include <Ubidots.h>

int button_pin   = D4;
int button_state = -1;

int rotation_pin   = A0;
int rotation_state = -1;

const char *WEBHOOK_NAME = "ubidots";
Ubidots ubidots("webhook", UBI_PARTICLE);

void setup() {
    pinMode( button_pin,   INPUT );
    pinMode( rotation_pin, INPUT );
}

void loop() {
    int button_state   = digitalRead( button_pin );
    int rotation_state = analogRead( rotation_pin );

    ubidots.add( "button_state",   button_state );
    ubidots.add( "rotation_state", rotation_state );

    ubidots.send(WEBHOOK_NAME, PUBLIC);

    delay( 1000 );
}

```

2. Building a Ubidots Dashboard

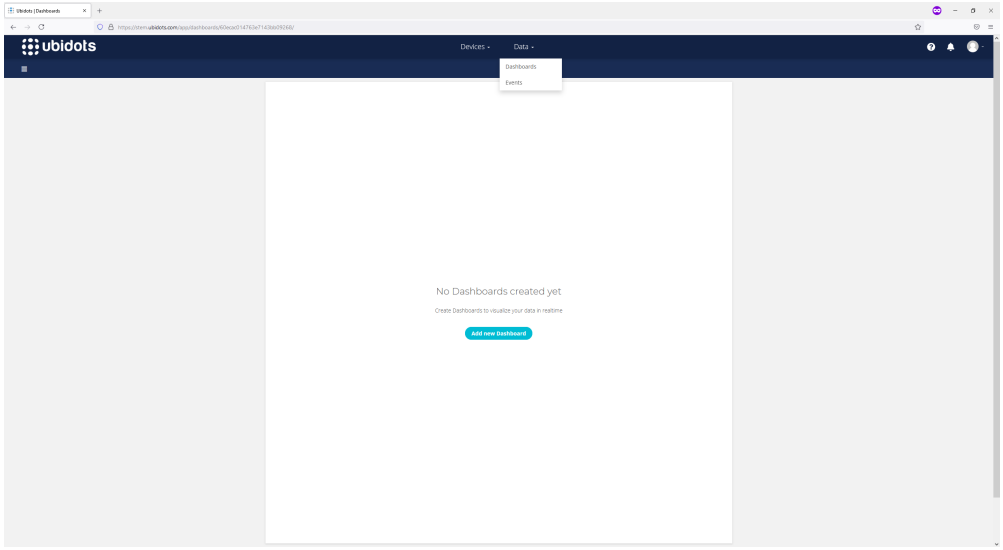
Now that Ubidots is receiving data from Particle, we need to tell it how we want the data displayed. Log in to Ubidots at `https://industrial.ubidots.com/accounts/signin/`. Click on the devices menu at the top of the page, and select the devices option. You should see a list of the devices that have sent data to this Ubidots account.



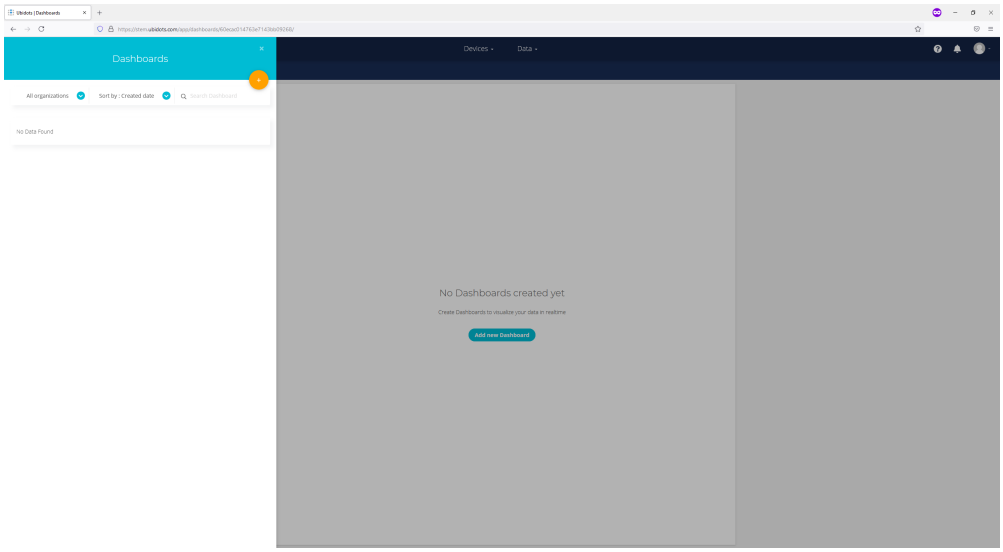
Click on the devices in the list to see what variables the device has sent to Ubidots.



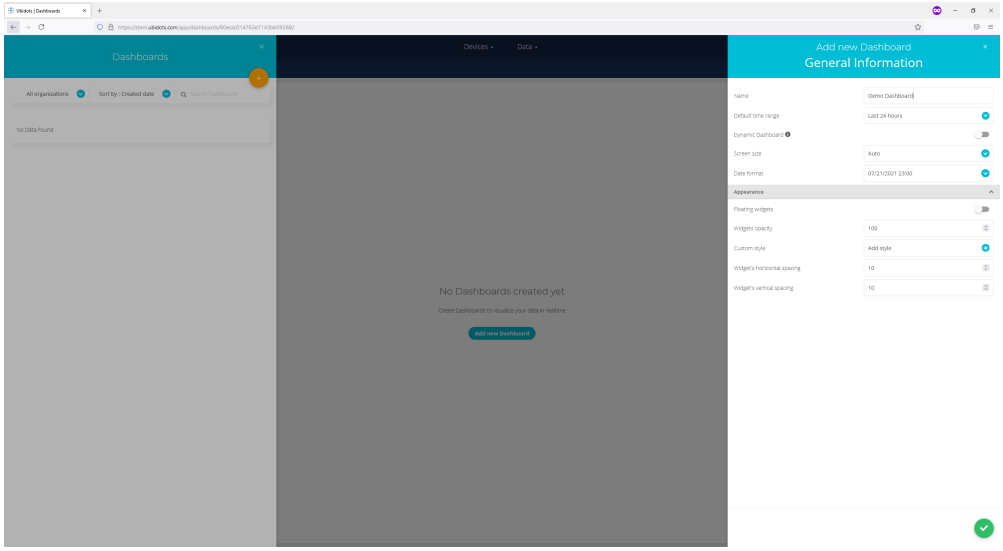
To begin building your dashboard, go to the top of the page, open the data menu, and select the dashboards option.



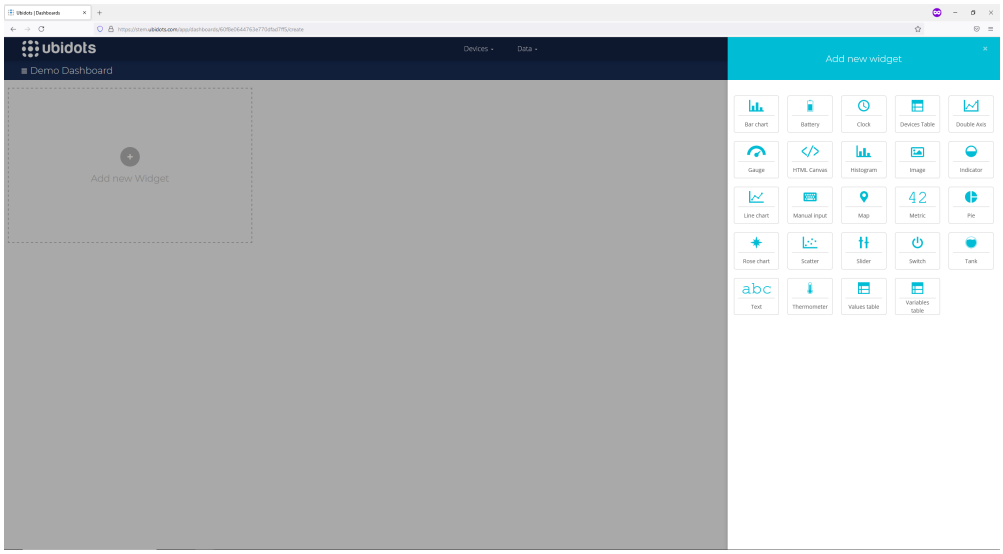
Click on the three-horizontal-bar menu button in the top left, and then the orange (+) button to create a new dashboard.



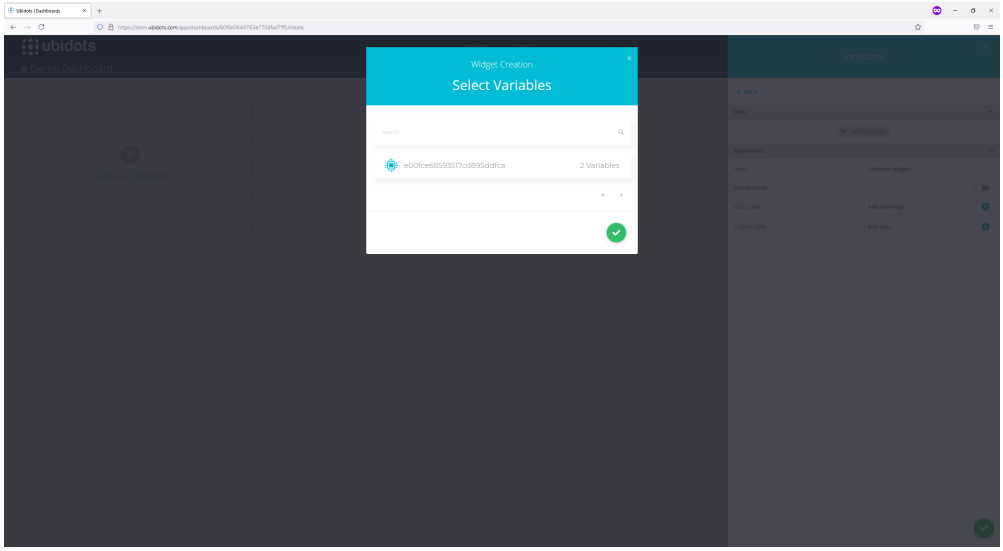
Name your dashboard and then click the green check mark button in the bottom right corner to create the dashboard.



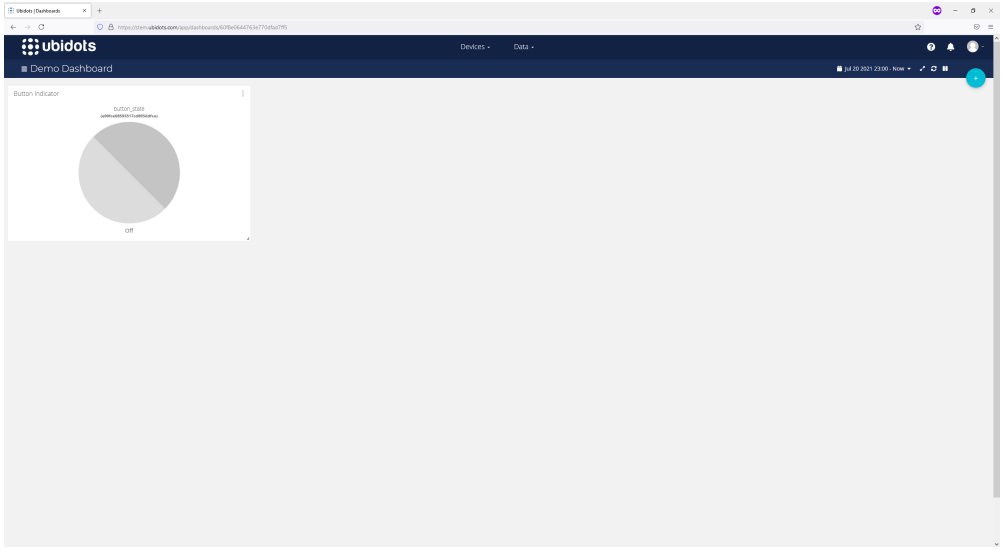
Add widgets to your dashboard by clicking the blue (+) button in the top right corner.



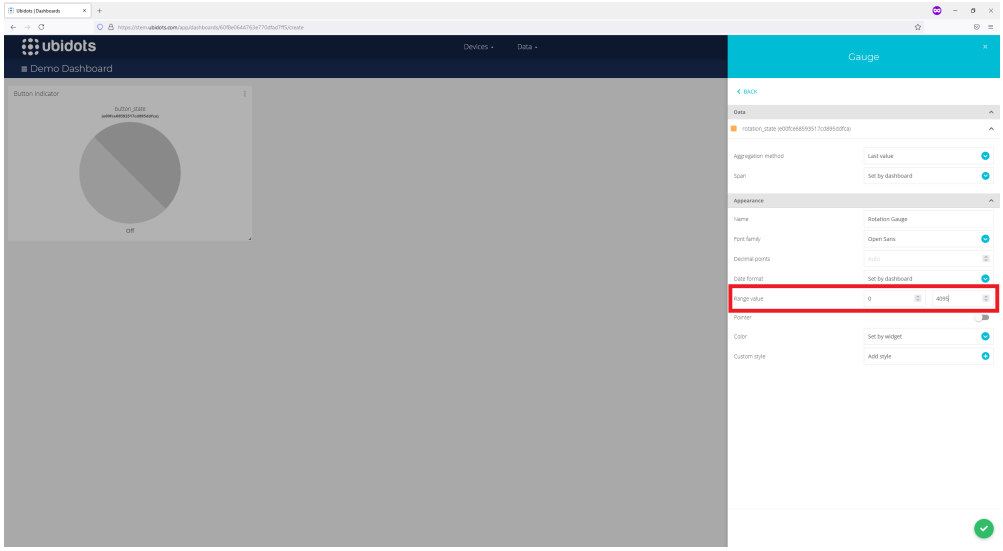
We will start with an "indicator" widget to show the state of our button. Click on the "Add Variables" button to bring up the variable search menu.



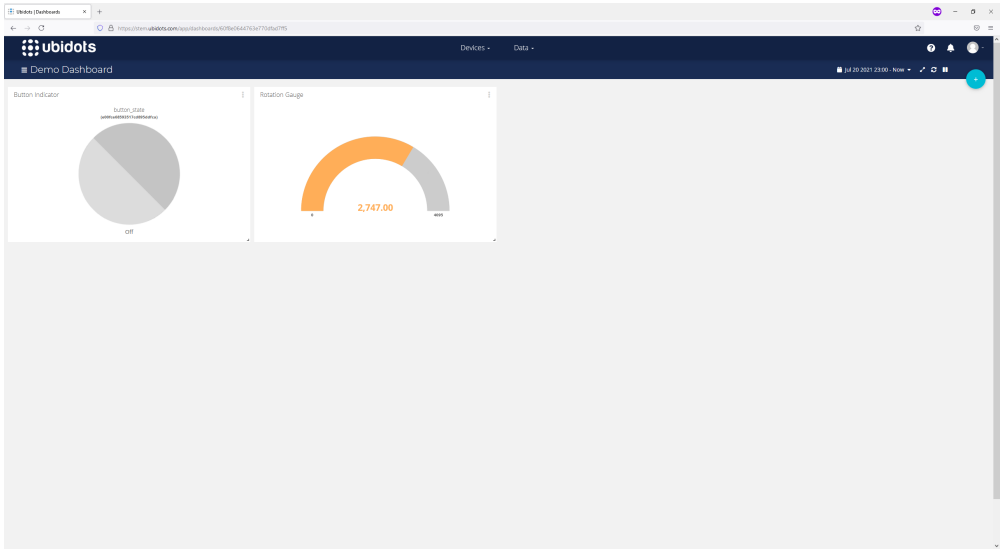
Click on your device, then the button variable, and then click the green check mark button. Name your widget and then click the green check mark in the bottom right corner.



Next, we will add a gauge widget to display the value from our rotation sensor. Use the same process to add your rotation sensor variable and name your widget. You will also want to edit the range value for this widget, because your rotation sensor will generate values between 0 and 4095.



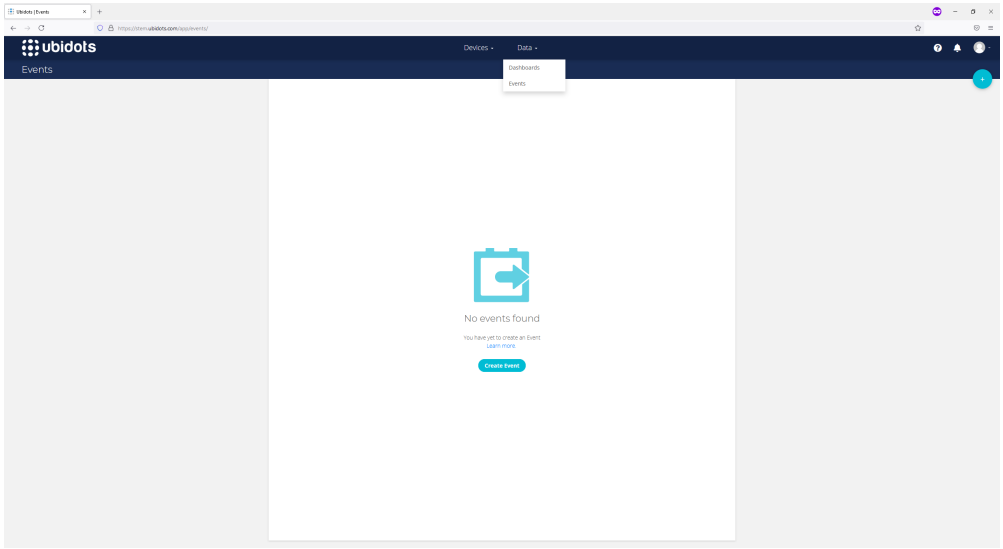
Turn your particle back on and reprogram it with the code from Section 1. As you push the button and turn the rotation sensor, you should see the widgets on your dashboard change.



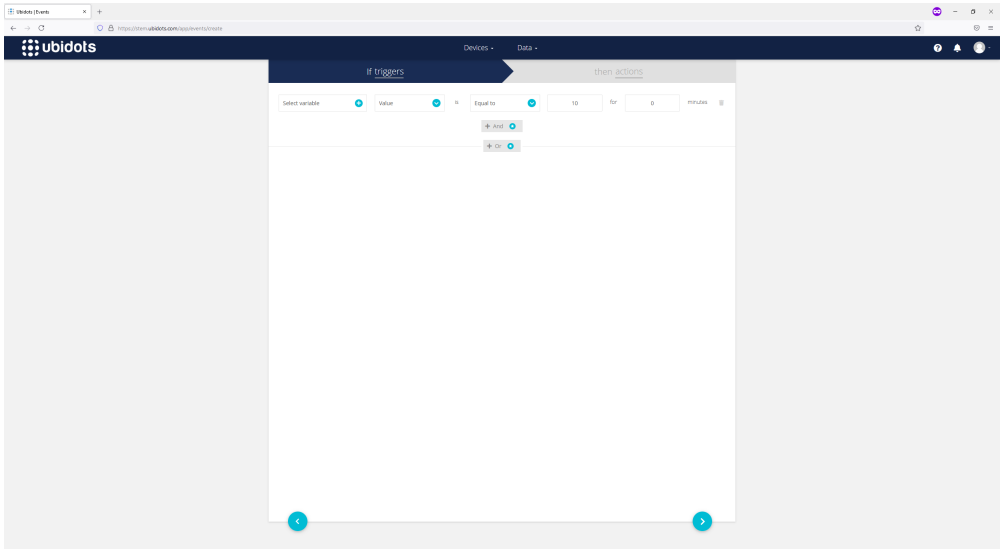
Other widgets you might want to explore for your project include the line chart (which is good for tracking the value of variables over time), the tank (which might be good for showing the value of a water or moisture sensor), and the thermometer (which could be good for displaying temperature values).

3. Creating Ubidots Events

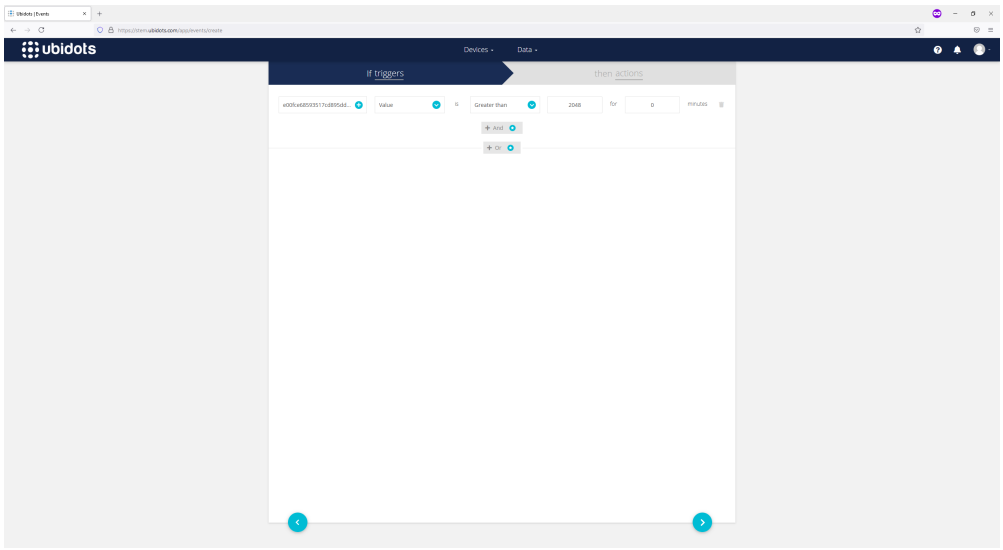
Ubidots events allow you to configure Ubidots to perform various actions when the values in your dashboard meet certain conditions. Let's have Ubidots send a text message whenever the rotation sensor goes over its halfway value of 2048. First, go to the top of the page and click on data, then events.



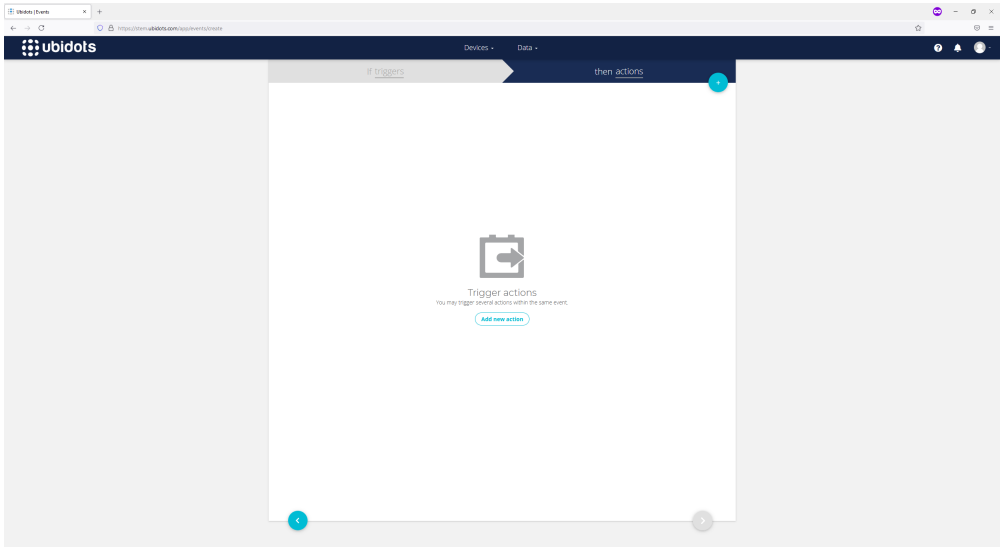
Click the blue (+) button in the upper right hand corner to create a new event.



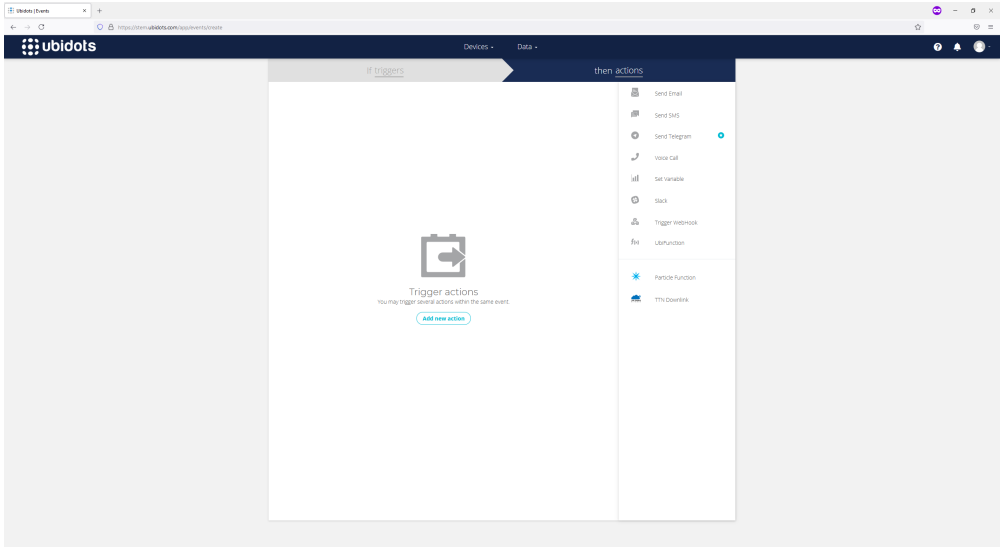
Click on the box that says "Select variable" to bring up the variable selection list, and select the variable for the state of your rotation sensor. Click on the box which says "Equal to" and select "Greater than" from the drop-down list that appears. Then change the value in the box next to that to 2048.



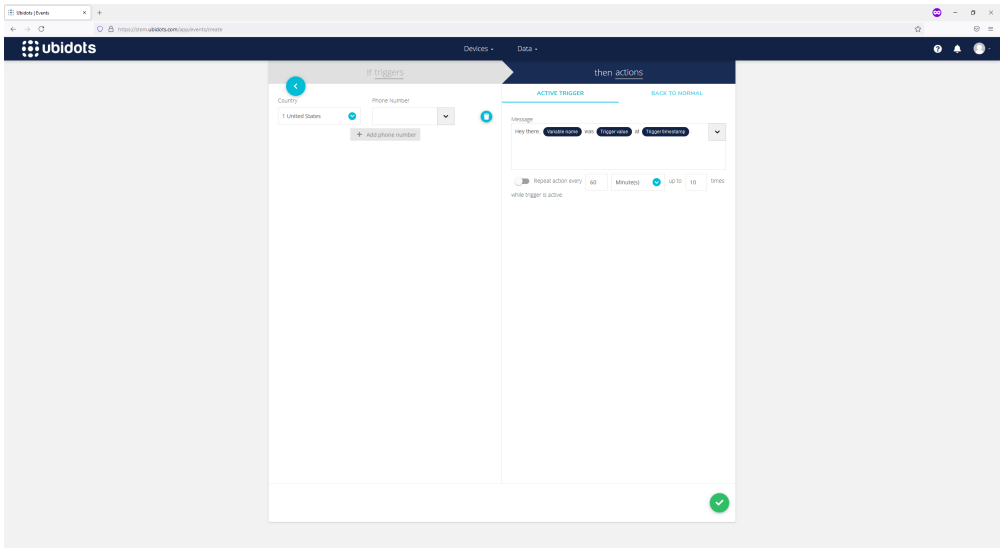
Click the blue arrow in the bottom right to move on to the next stage of event setup.



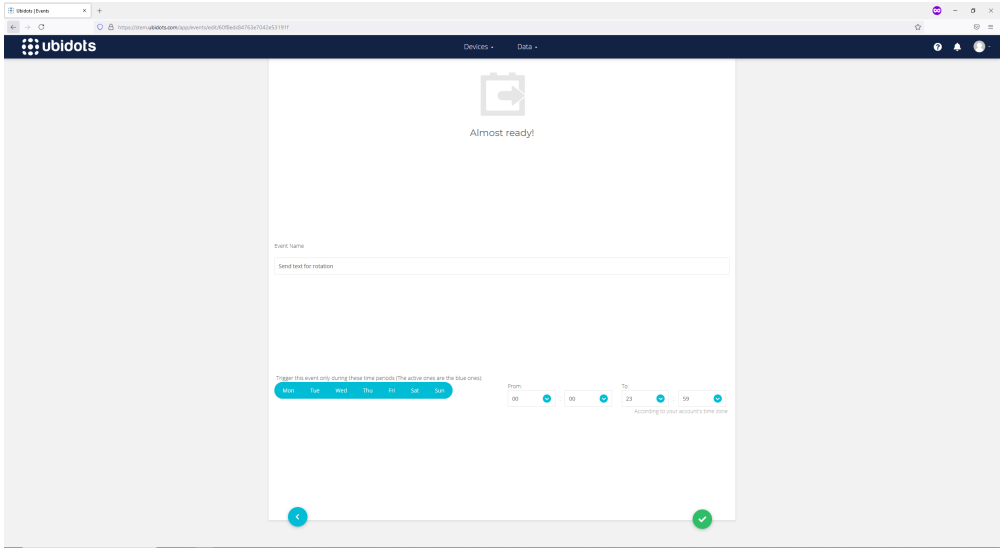
Click the blue (+) button near the top of the screen to add a new action for Ubidots to perform when the trigger condition is met.



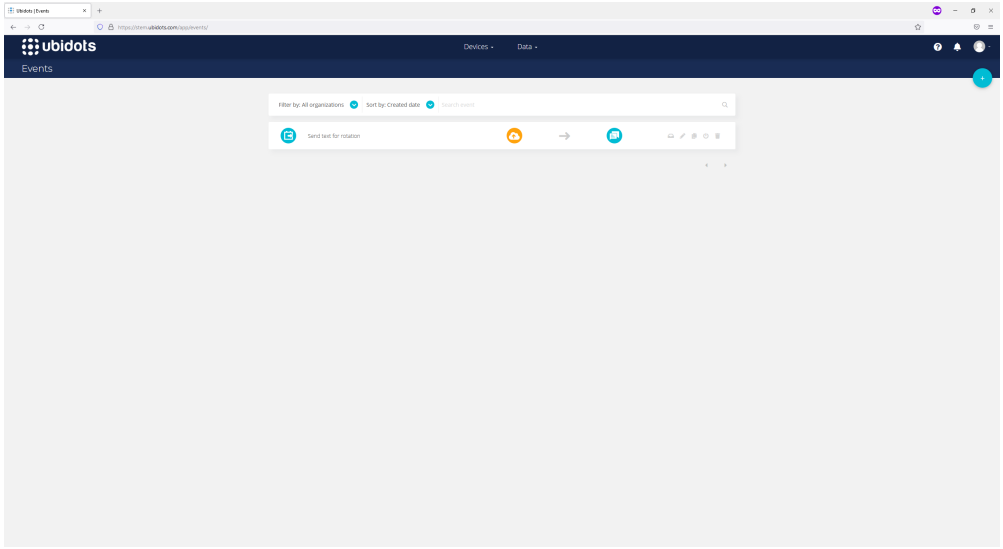
Select "Send SMS" from the list on the right side of the screen.



Enter your phone number in the box on the left hand side of the screen, then click the green check mark in the bottom right corner of the screen. You could add additional actions to the event at this stage, but for now we'll click the blue arrow in the bottom right to continue with the event creation process.



Name your event, and optionally restrict the times when it can be activated, then click the green check mark in the bottom right corner to finish creating the event.



You are now ready to test the event. With your Particle Argon running the code from Section 1, spin the rotation sensor back and forth, and see that you get a text message each time the value changes from less than 2048 to more than 2048.