Build the ESP32 Modular Project

Contents

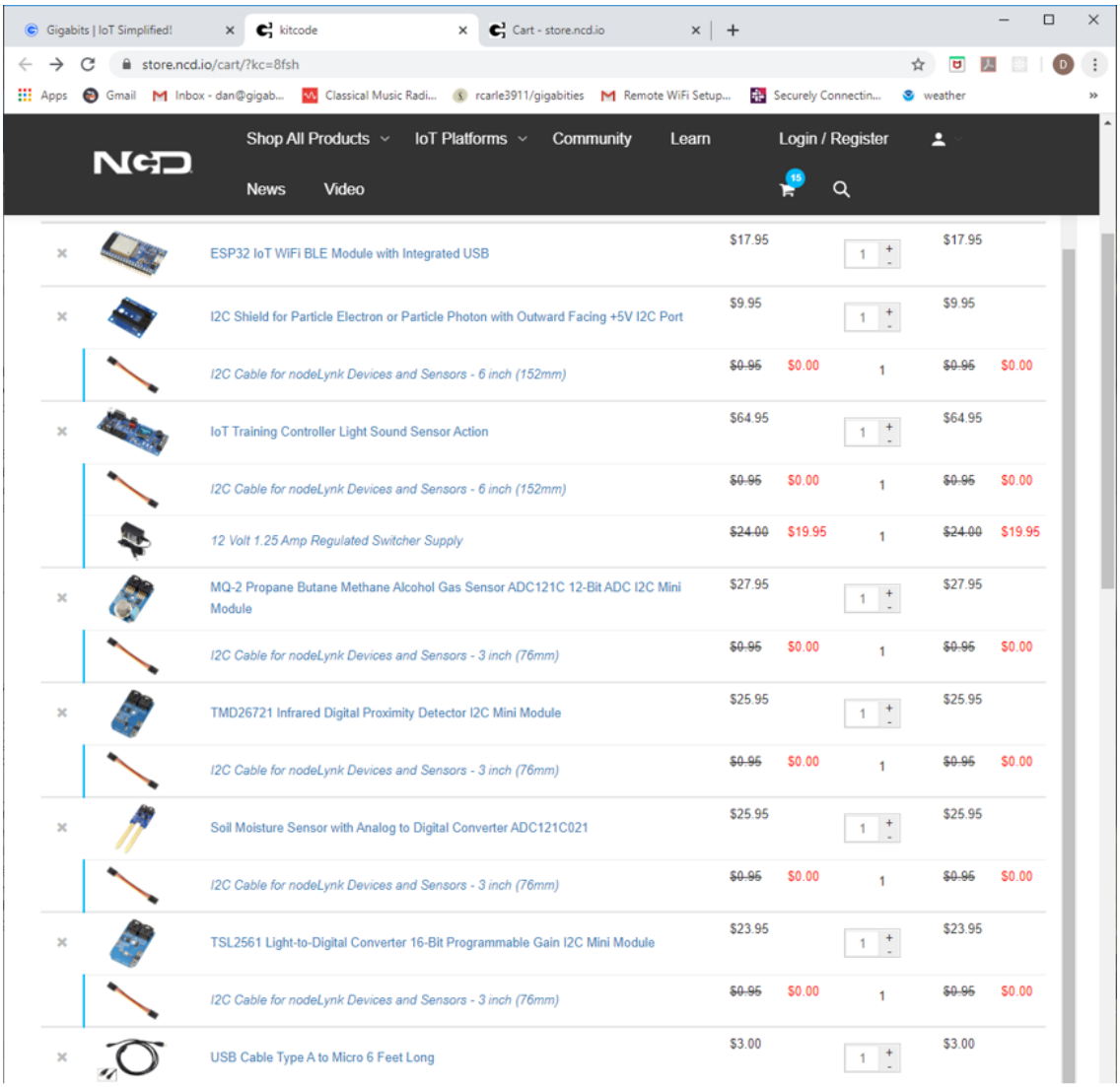
Introduction

The Gigabits IoT Platform is a backend that makes it easy to acquire, display, store and analyze data from projects. Gigabits IoT Platform's ability to combine a user's data with remote datasets makes it stand out.

This project shows how to use the Gigabits IoT Platform with a several modular sensors from National Control Devices (NCD) produce a simple plant monitor or weather station. It illustrates the use of 8 physical sensors and one button.

Assembling the Kit

Open a web browser to our web page at gigabits.io. Scroll down to the bottom of the page and click “Click here to get your Enterprise Sandbox Kit from NCD”



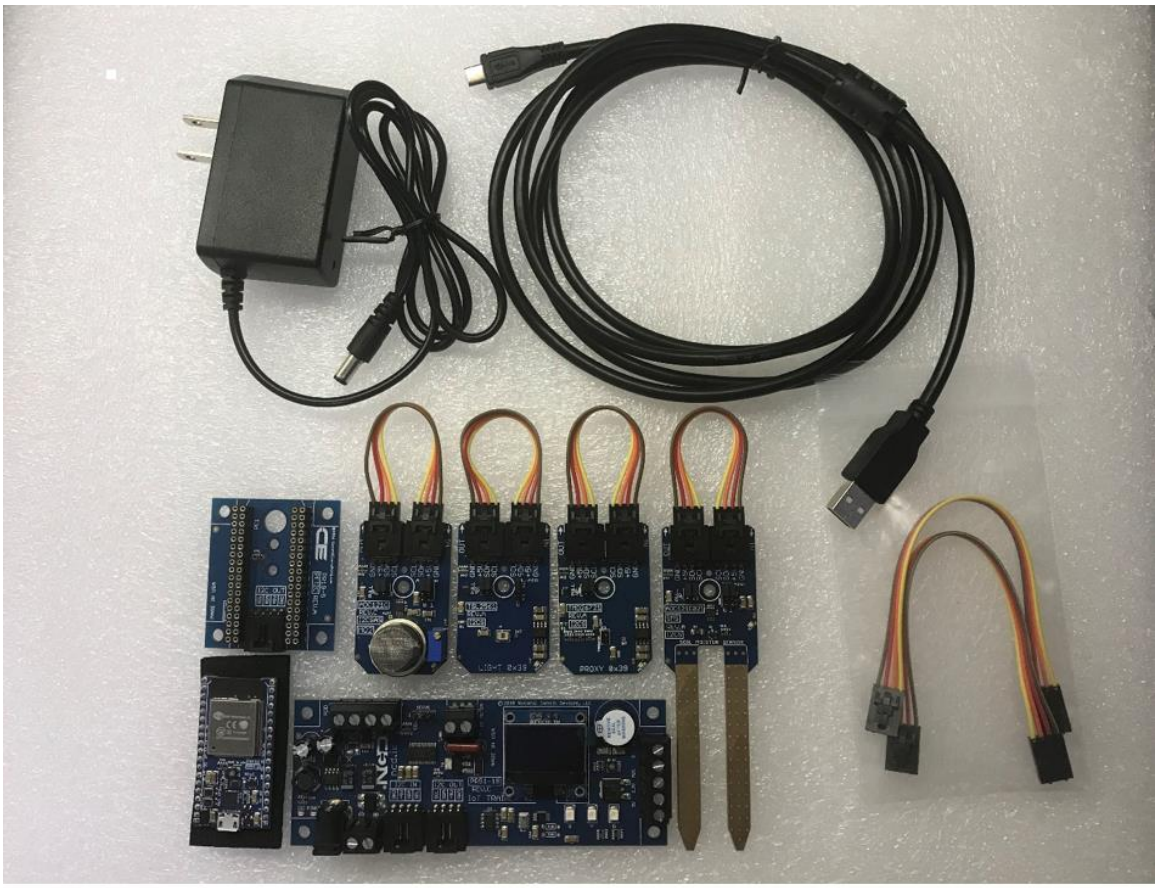
Modular Figure 1

There are several versions of ESP32 on the market. The one in the kit has been modified by NCD to support the NCD sensors and actuators. That means ESP32's from other sources may not work with this kit.

Kit Step 1: At a minimum, you will need an "ESP32 IoT BLE Module with Integrated USB", a "I2C Shield for Particle Electron or Particle Photon with Outward Facing +5V I2C Port" and one of the remaining boards. We recommend the "IoT Training Controller Light Sound Sensor Action" board with the 12 Volt 1.25 Amp Regulated Switcher Supply". This "Training board" has several types of sensors and actuators. It was designed to introduce new users to IoT. You may choose as many of the remaining boards as you like.

Kit Step 2: Proceed to checkout and order the kit.

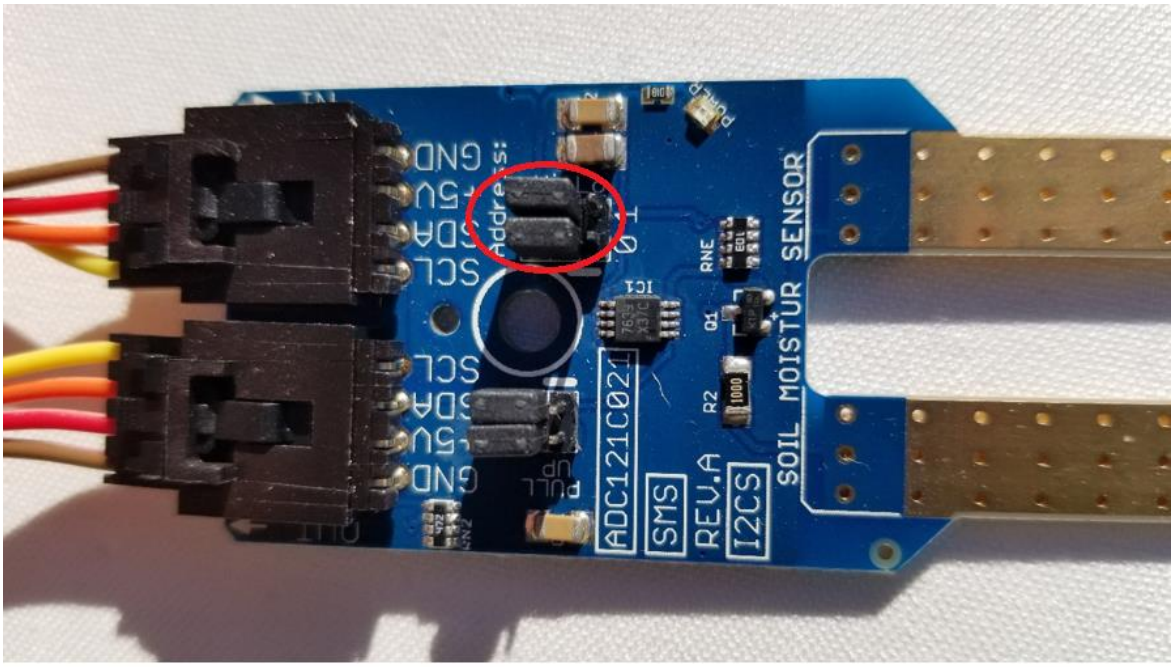
Kit Step 3: When the kit arrives, unpack it and make sure that you got everything you ordered. Some of the I2C colorful ribbon cables may be attached to the sensor boards rather than being packaged separately.



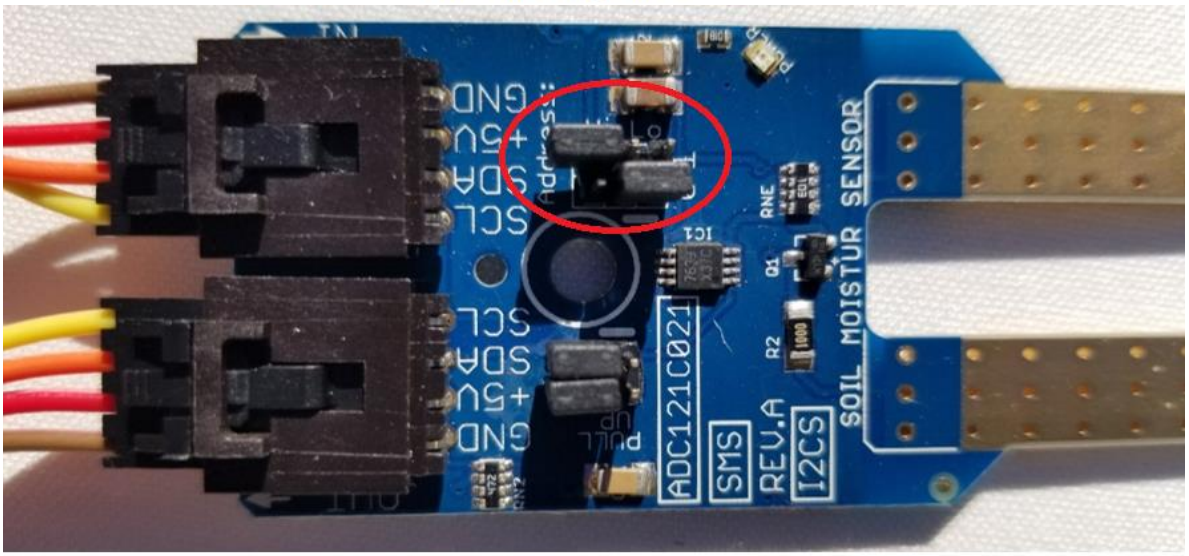
Modular Figure 2

Kit Step 4: We need to set an address jumper on the gas sensor, another one on the soil moisture sensor and one on the light sensor. These prevent multiple sensors from attempting to respond to the same address.

Kit Step 4a: Let's start with the soil moisture sensor (the one with the two prongs that you push into dirt). There are six pins labelled "Address:". The pins are arranged in two columns of 3 pins each. The two columns of pins are labeled "0" and "1" as shown in Figure 3. Leave the jumper from the column of pins labelled "1" alone. Move the jumper in column 0 so that it covers the two pins that are furthest from "Address:".

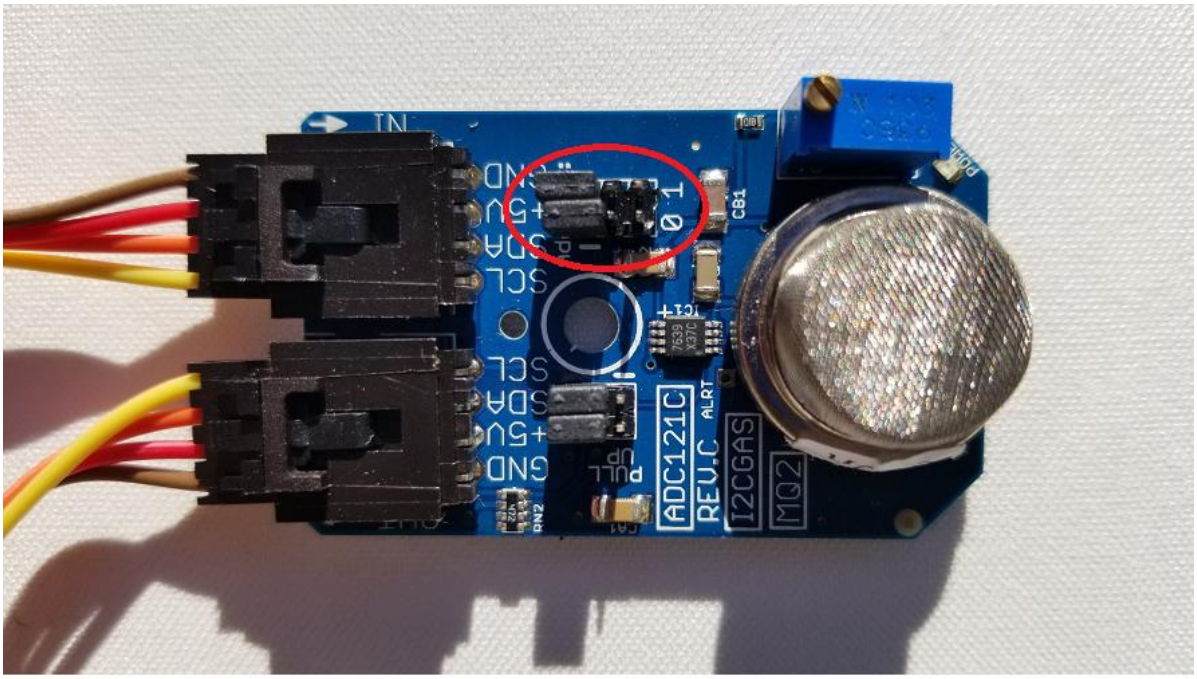


Modular Figure 3



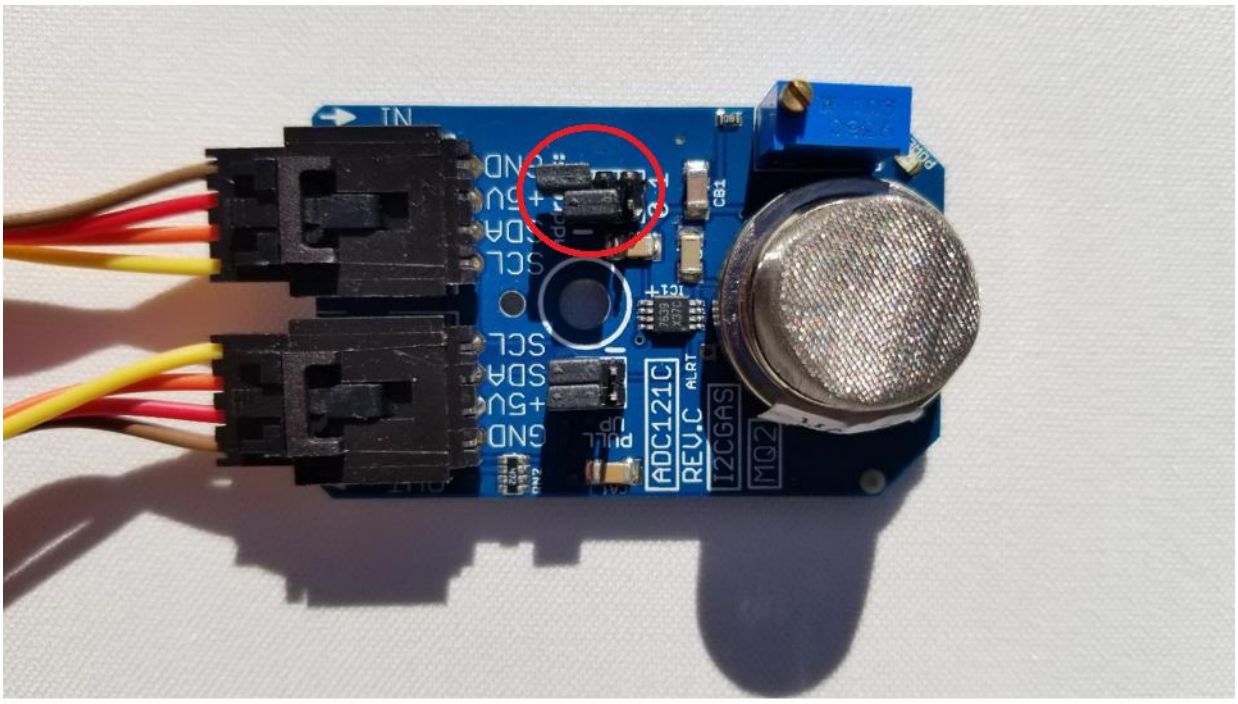
Modular Figure 4

Kit Step 4b: We do something that's almost the same to the Gas sensor.



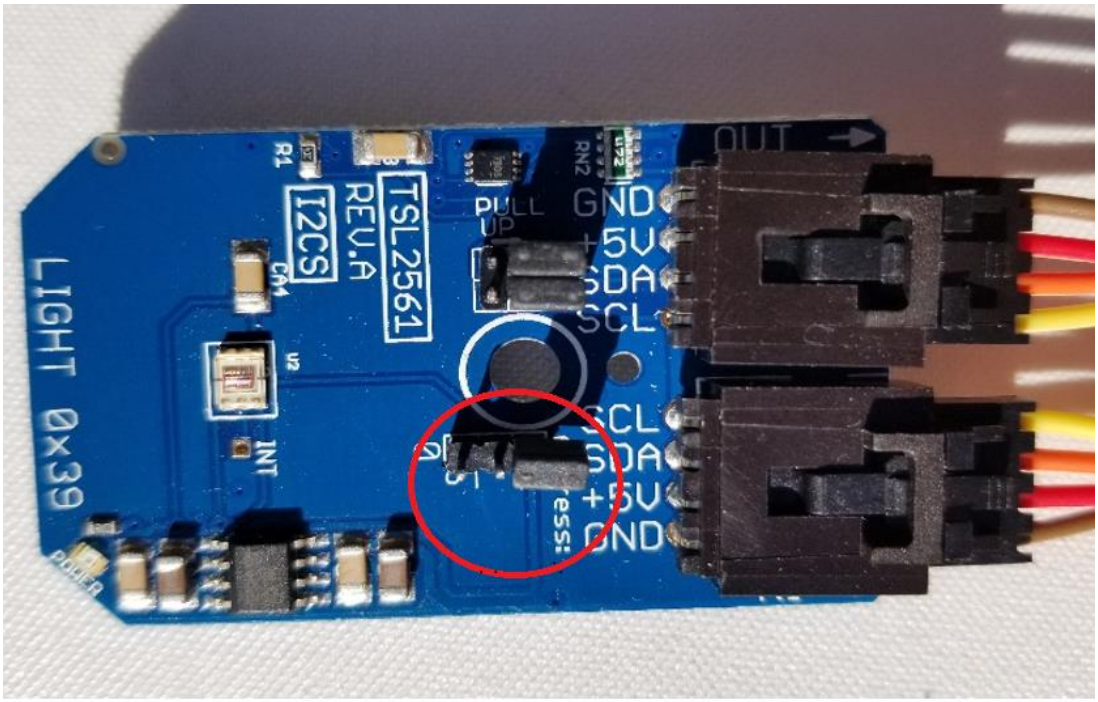
Modular Figure 5

As we did with the moisture sensor, leave the jumper in the column labelled “1” alone. Connect the jumper in the column labelled “0” so that it covers the two pins closest to the “Address:” text.



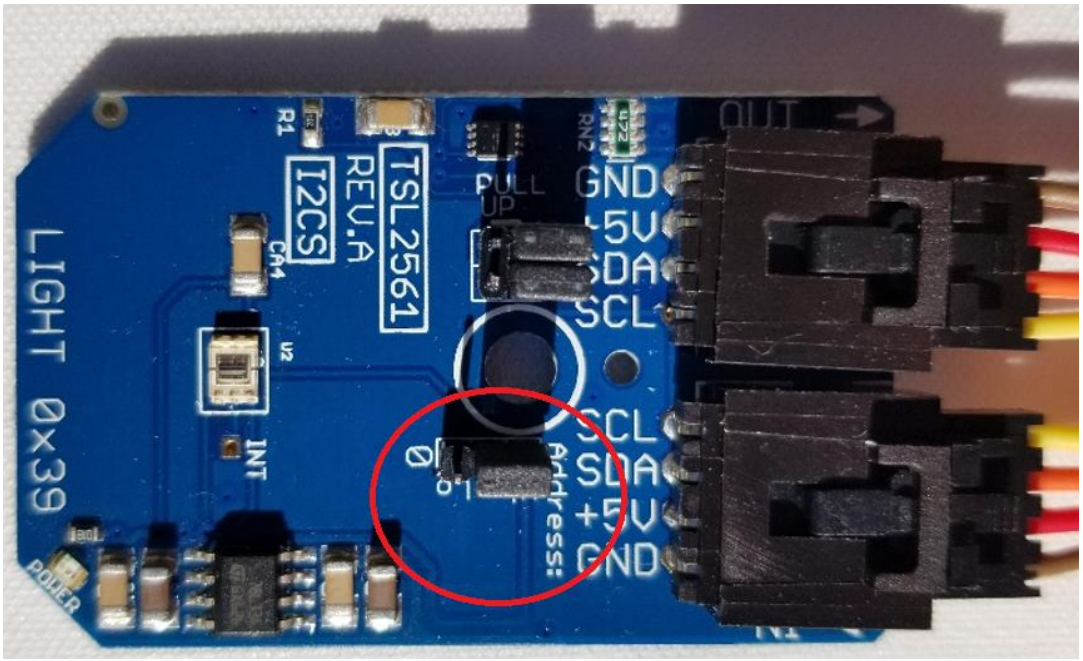
Modular Figure 6

Kit Step 4c: The TSL2561 Light sensor also needs a jumper. It has only one column of 3 pins.



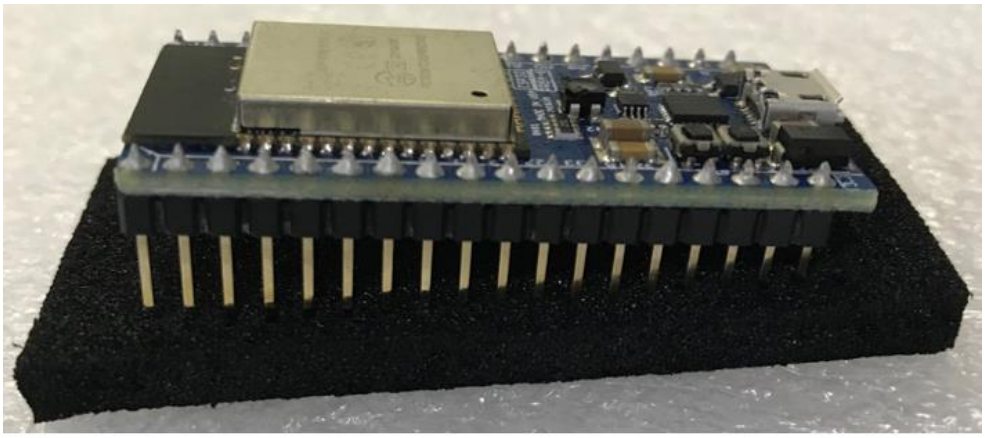
Modular Figure 7

Move the jumper so that it covers the two pins closest to “Address”.



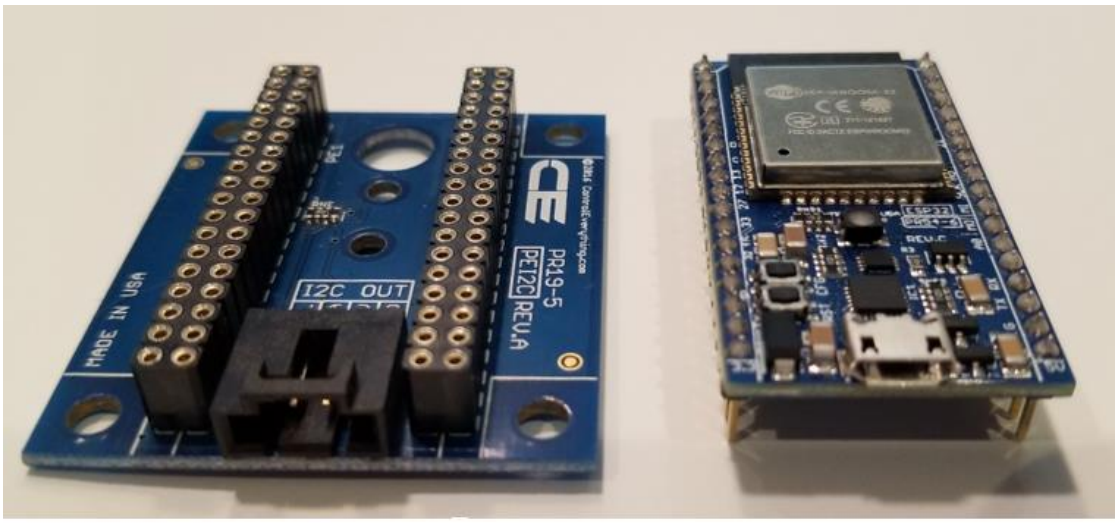
Modular Figure 8

Kit Step 5: Now that the boards are prepared, let’s continue by assembling the ESP32. Remove the piece of black, conductive foam from the ESP32’s pins.

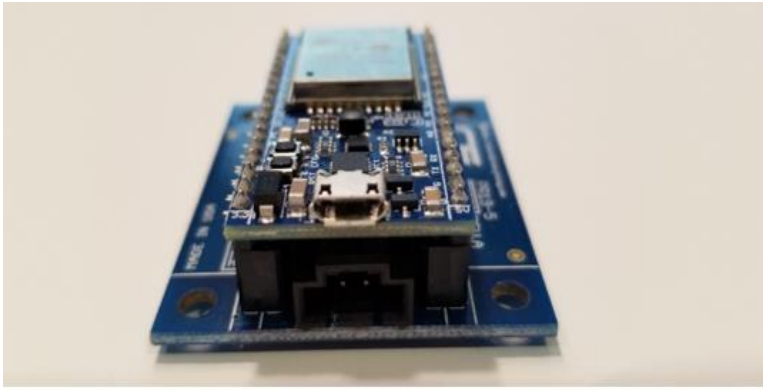


Modular Figure 9

Kit Step 6: Press the ESP32 into its adaptor board. The USB micro connector on the ESP32 should be above the I2C connector on the adaptor board. Be firm, but careful not to bend any pins.

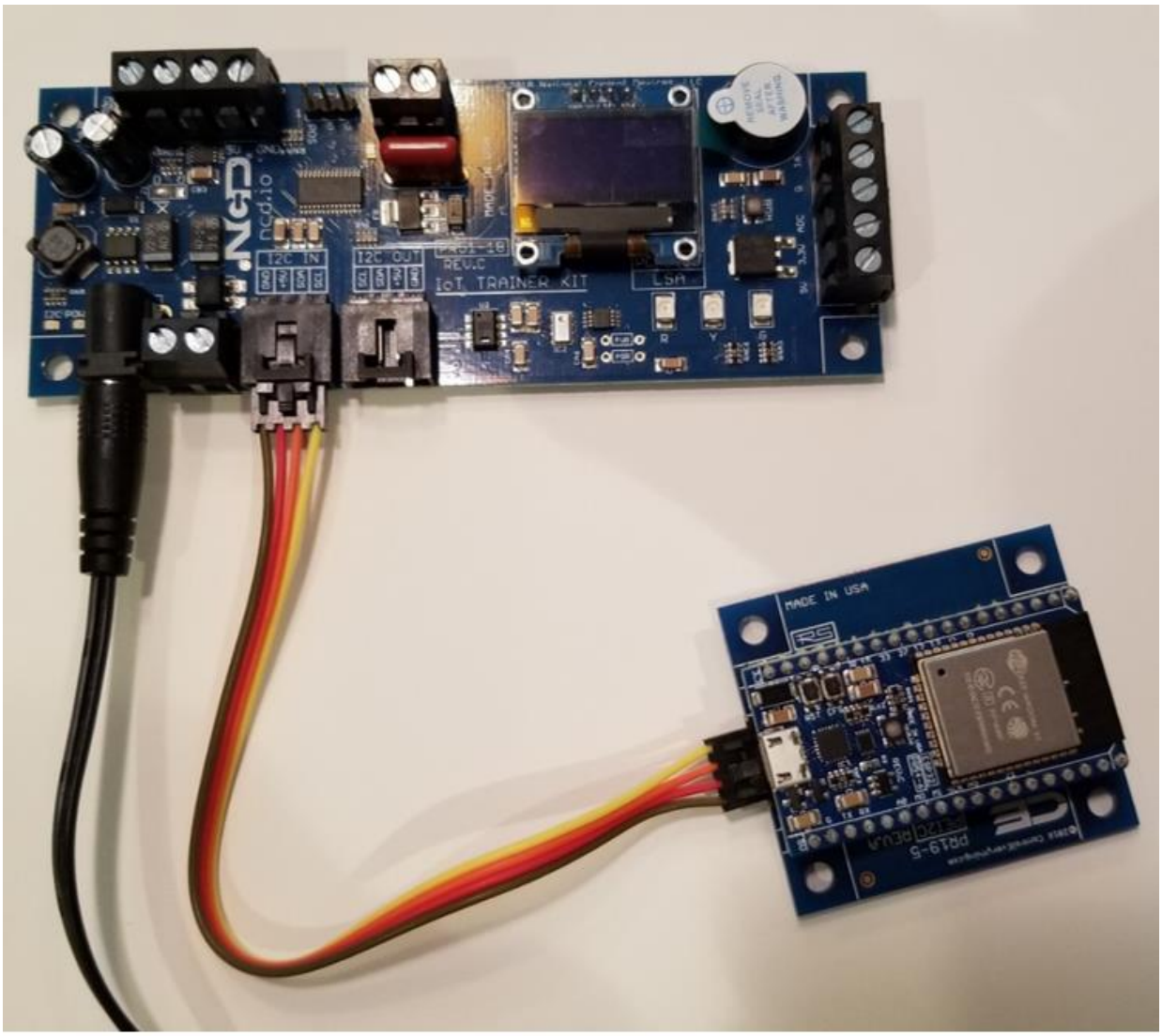


Modular Figure 10



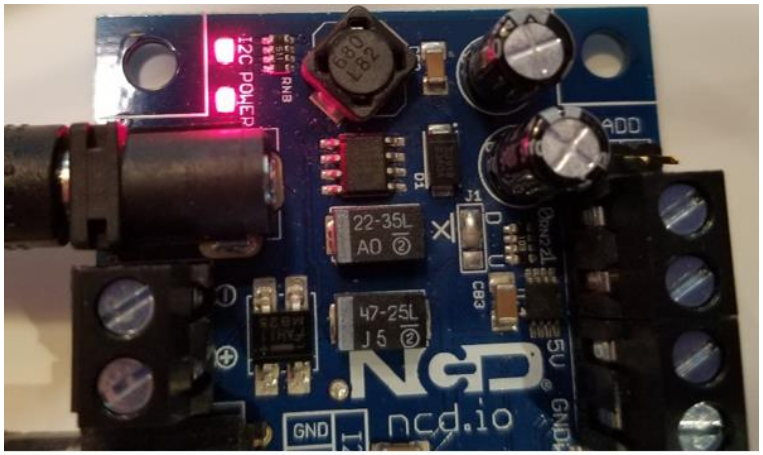
Modular Figure 11

Kit Step 7: Use one of the multicolored I2C cables to connect the ESP32 assembly into the training board. Plug the power supply's cable into the Training board. Then plug the power supply into a wall outlet. There are two LEDs next to the Training board's power supply connector. One LED on the training board should light up.

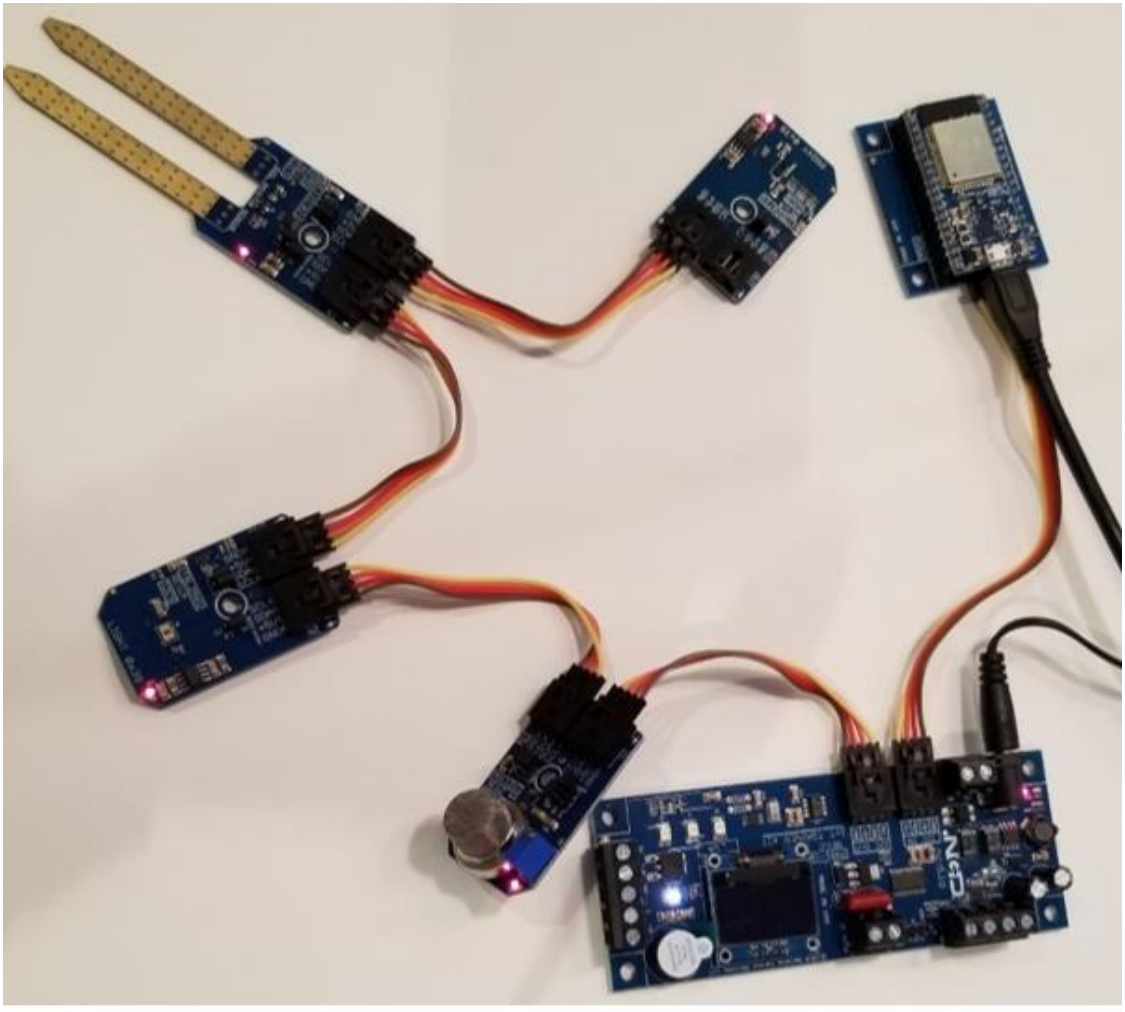


Modular Figure 12

Kit Step 8: Using the multicolored I2C cables, connect the rest of the sensors in a chain. For the sake of consistency, connect the brown wire in each ribbon cable to GND on the board. The order in which boards are chained together should not matter. When this step is complete, two LEDs on the training board and one LED on each of the remaining boards should be lit.

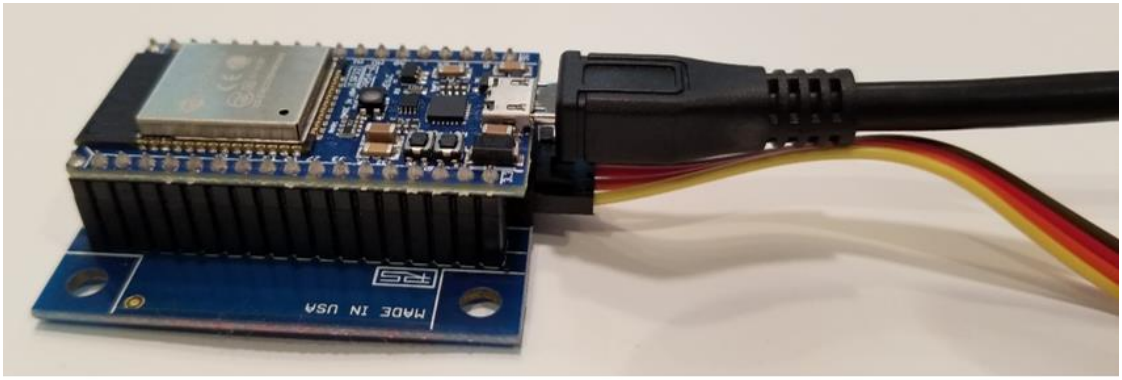


Modular Figure 13



Modular Figure 14

Kit Step 9: Connect the USB A to Micro B cable between the ESP32 Development board and your PC or laptop. The ESP32's USB connector appears to be fairly delicate. Be gentle.



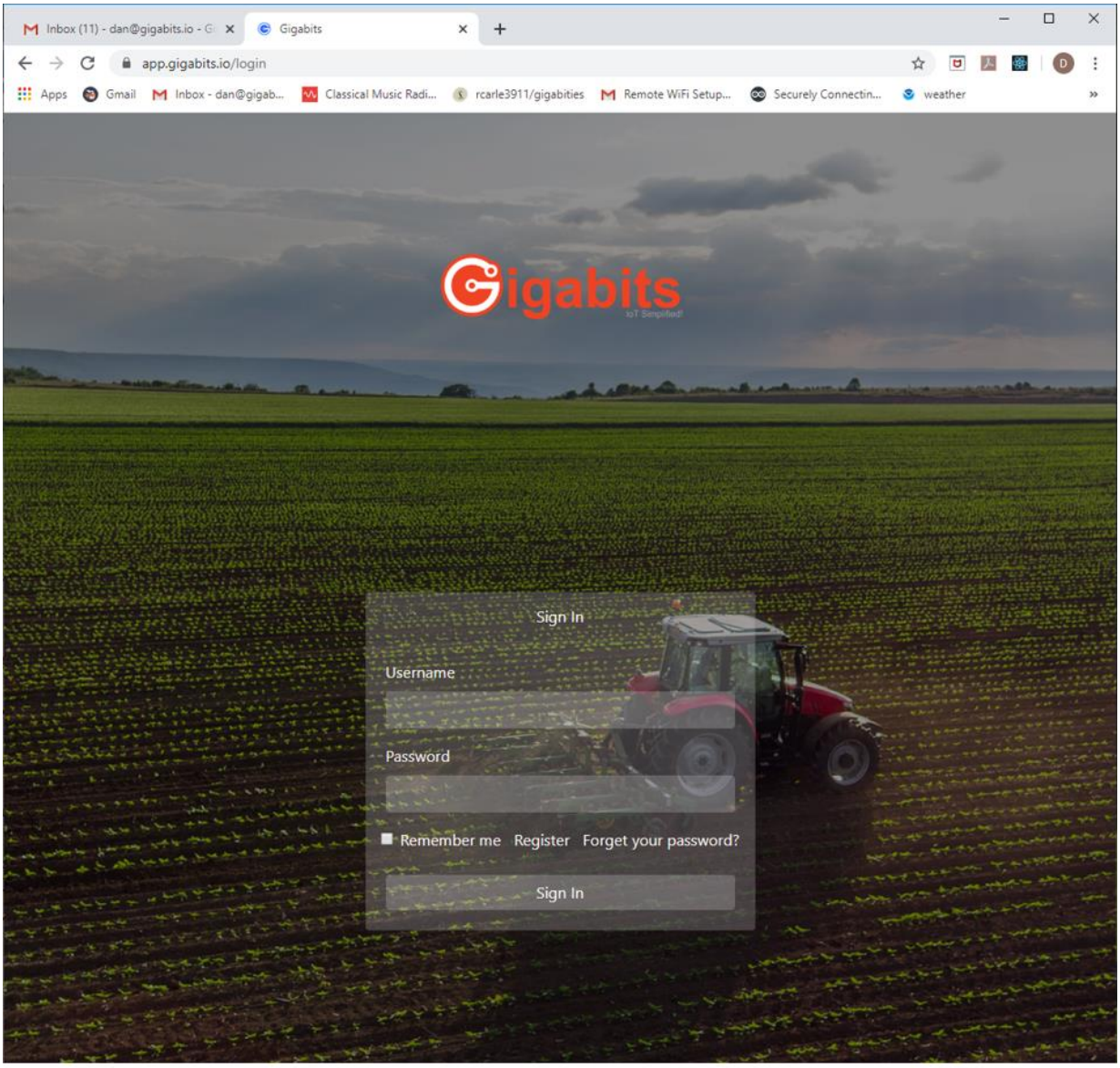
Modular Figure 15

Modular Figure 16

Kit Step 10: The kit has been assembled. Proceed to the next step to install the software.

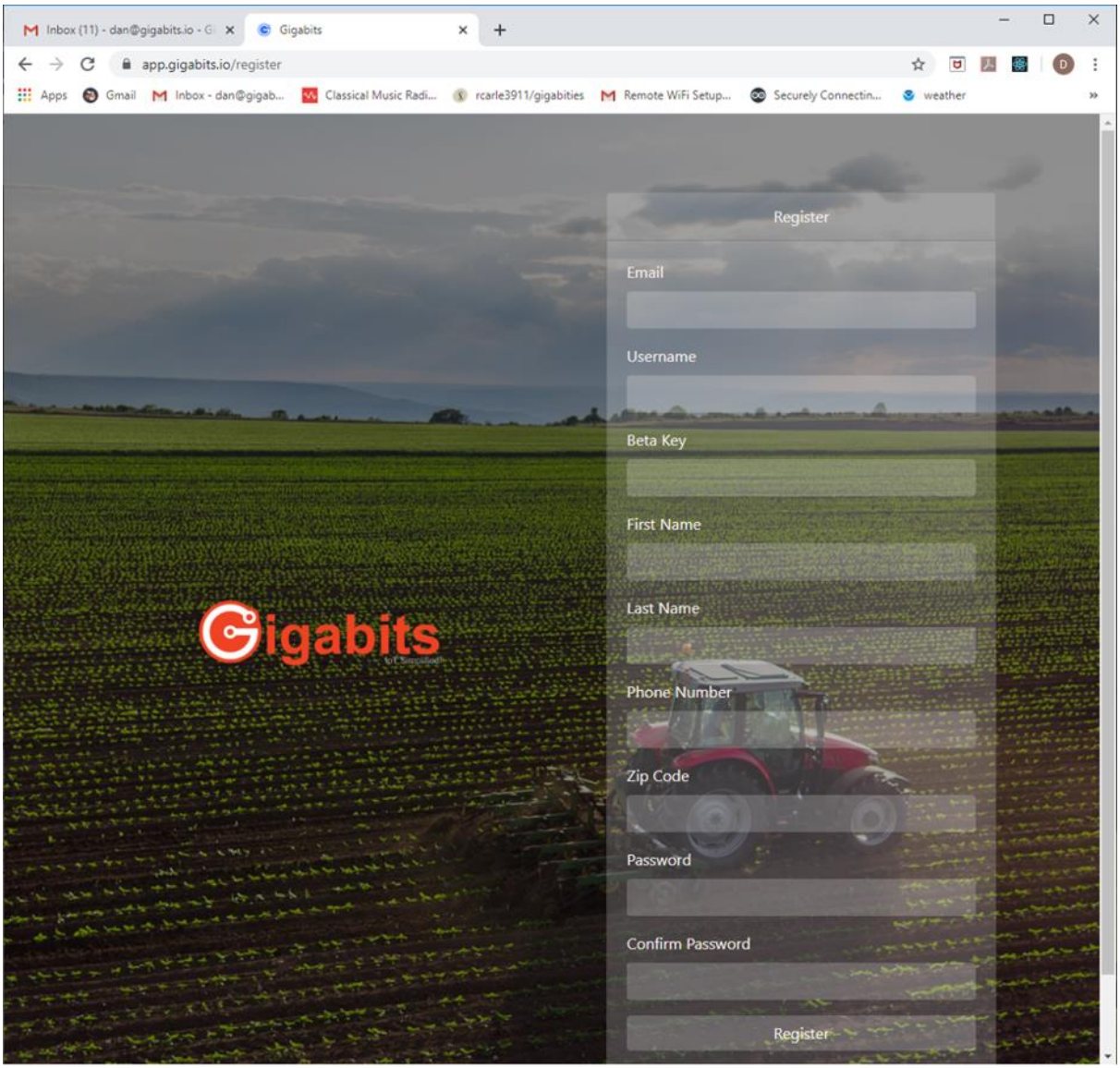
Installing Gigabits ESP32 software

Software Step 1: Open a web browser and go to “app.gigabits.io”.



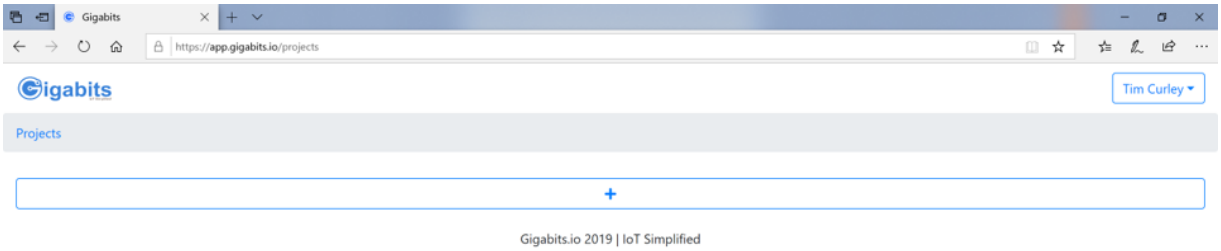
Software Figure 1

Software Step 2: If you haven’t registered with Gigabits yet, use your Beta Key to register now.



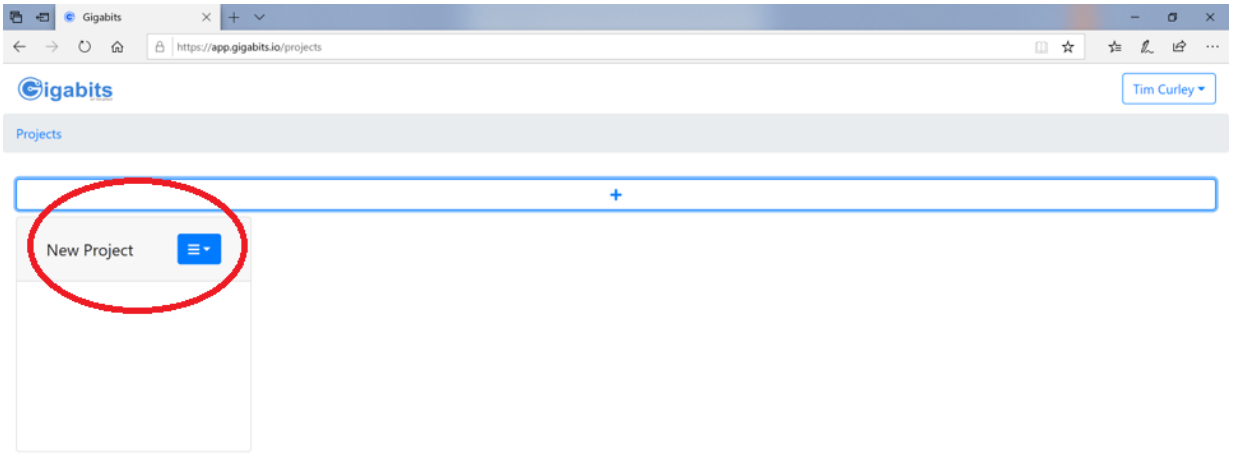
Sofware Figure 2

Software Step 3: Once you've registered, go back to app.gigabits.io and sign in.



Software Figure 3

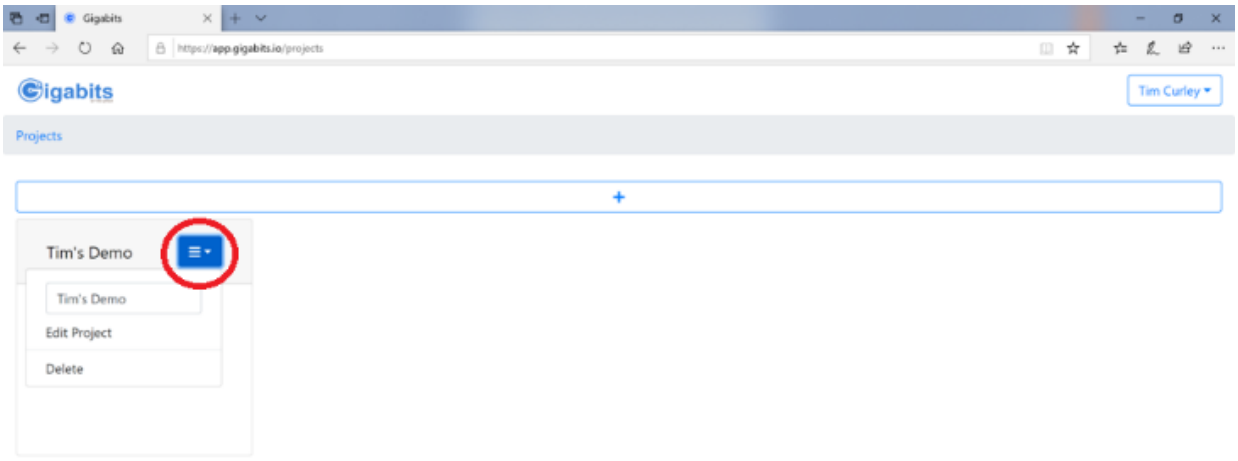
Software Step 4: Create a new project by selecting the plus sign. This will create a project named "New Project"



Gigabits.io 2019 | IoT Simplified

Software Figure 4

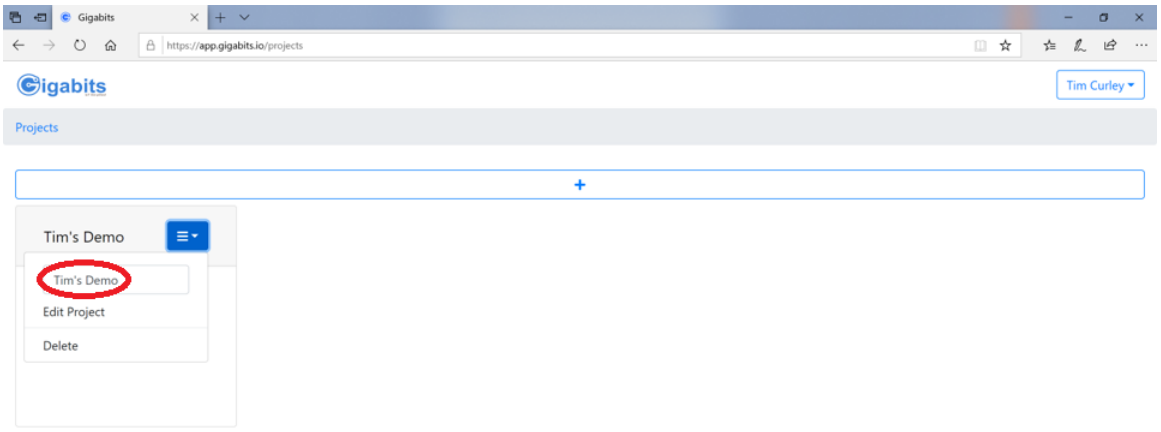
Software Step 5: Select the blue square that's next to your project's name and give your project a nice name by replacing "New Project".



Gigabits.io 2019 | IoT Simplified

Software Figure 5

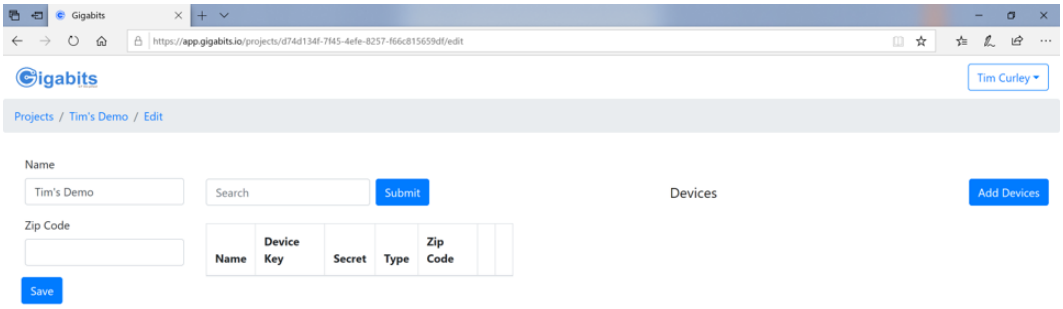
Software Step 6: Select "Edit Project" after pressing the project's blue square. This should show you the page used to edit projects.



Gigabits.io 2019 | IoT Simplified

Software Figure 6

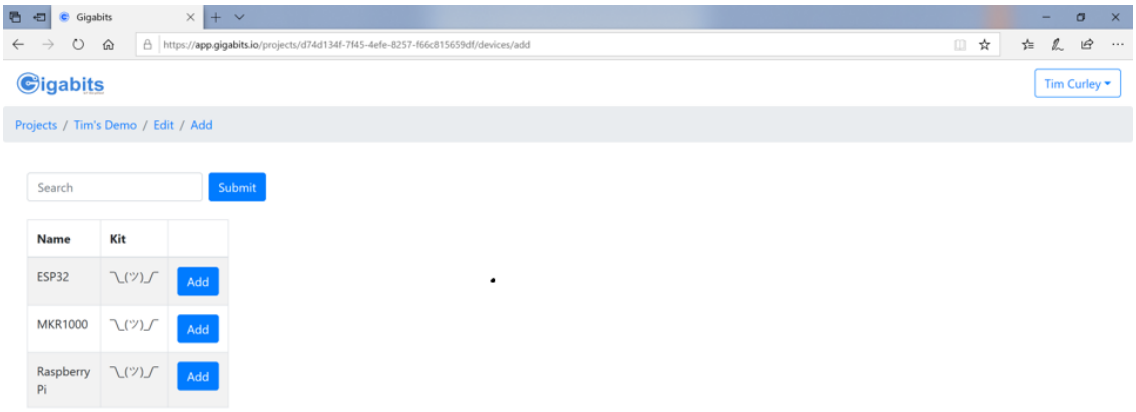
Software Step 7: Every project should have exactly one device (microcontroller). Press the blue "Add Devices" button on the far right. If you can't see that button, make your browser window wider.



Gigabits.io 2019 | IoT Simplified

Software Figure 7

Software Step 8: Add an ESP32 device for your project.

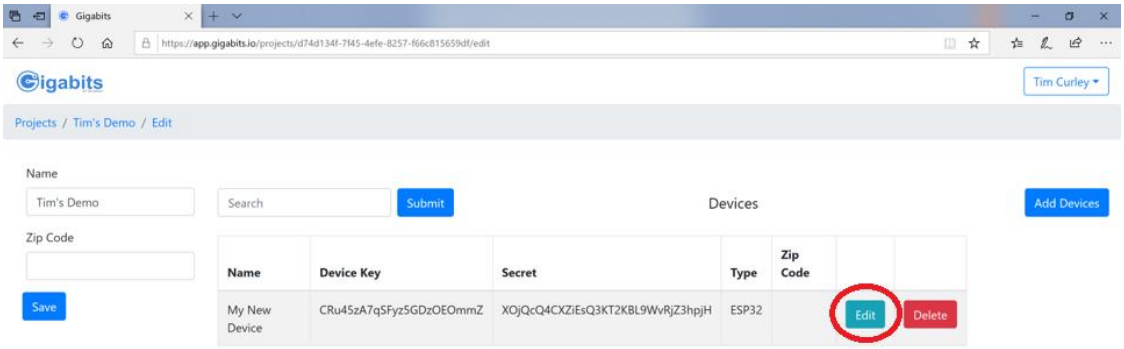


Gigabits.io 2019 | IoT Simplified

Software Figure 8

Software Step 9: If you see a mostly blank screen, try clicking on “Gigabits” in the upper left. Then select your project’s blue square and select “Edit Project”.

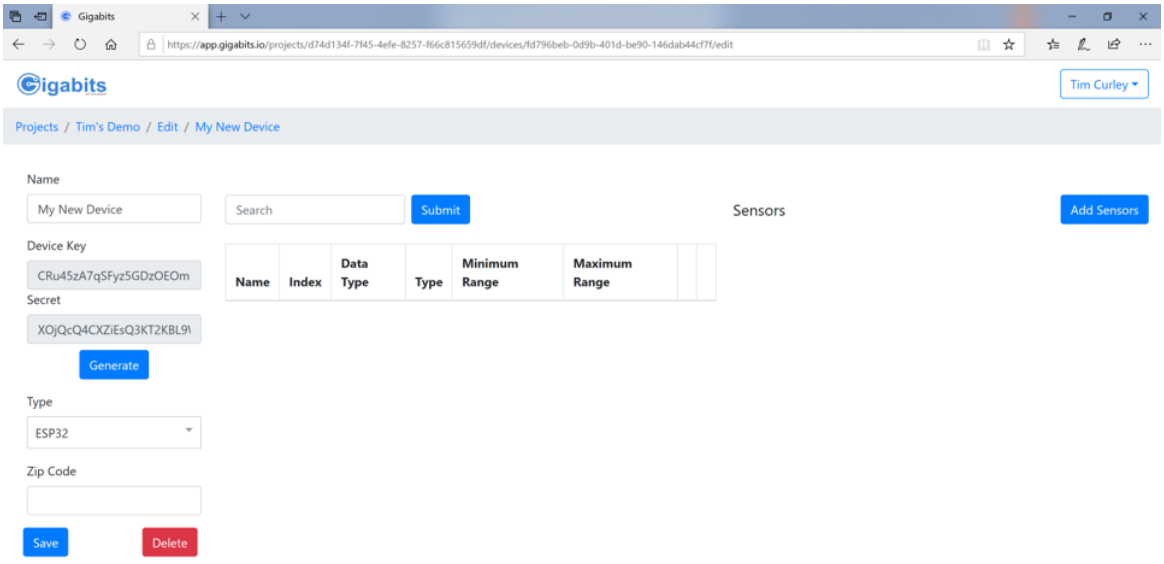
Software Step 10: You should now see the page used to edit a project. It should have a device.



Gigabits.io 2019 | IoT Simplified

Software Figure 9

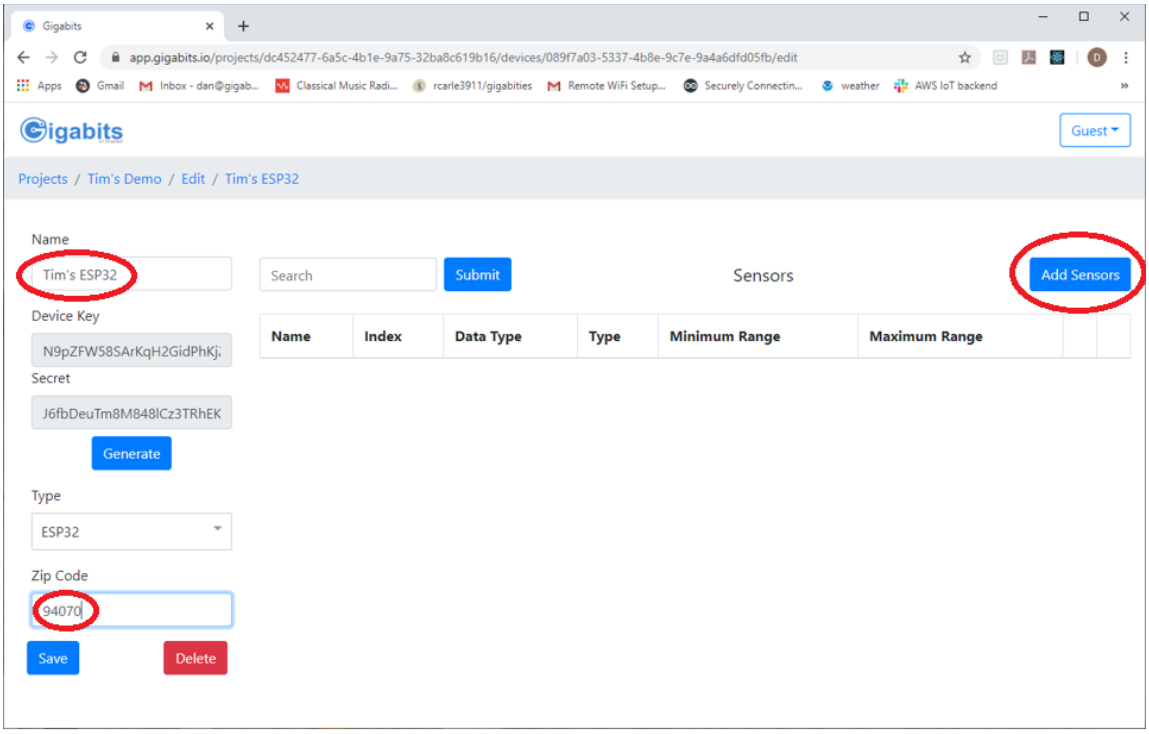
Software Step 11: Click the “Edit” button in your device’s row. This will take you to the page used to edit devices.



Gigabits.io 2019 | IoT Simplified

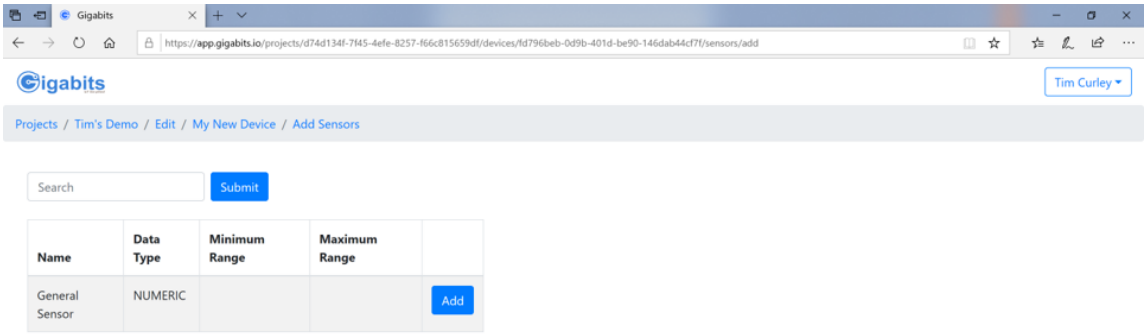
Software Figure 10

Software Step 12: Give your device a descriptive name by replacing the default ("My New Device"). Later on, we'll want to display weather data from around the country. Fill in the Zip Code that specifies which weather you'd like to see.



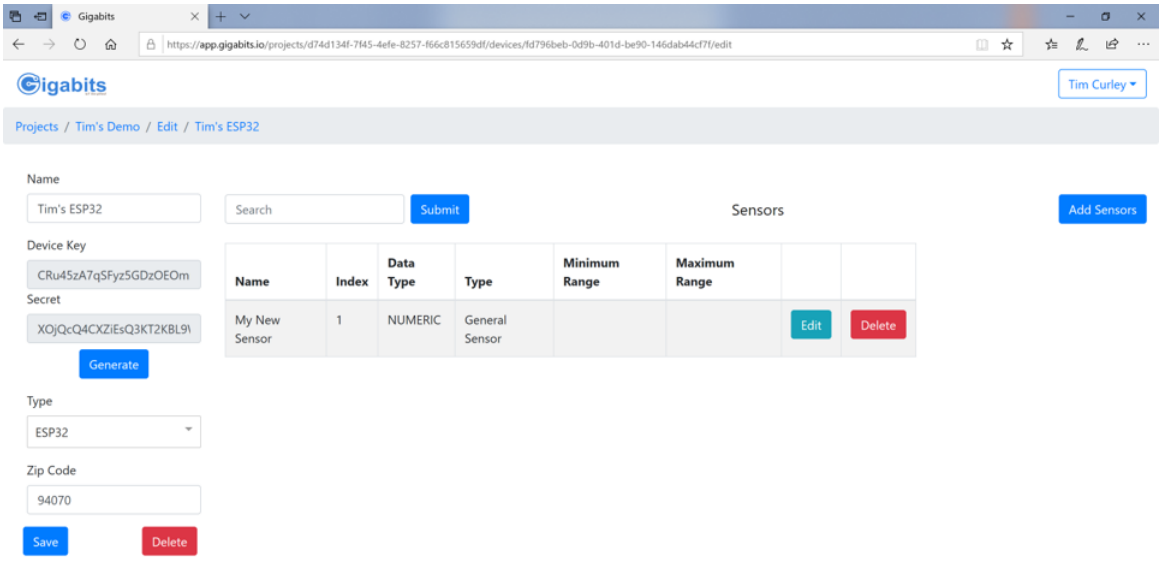
Software Figure 11

Software Step 13: Add a sensor to your device by clicking on the "Add Sensors" button. This will take you to a page that describes the sensor being added.



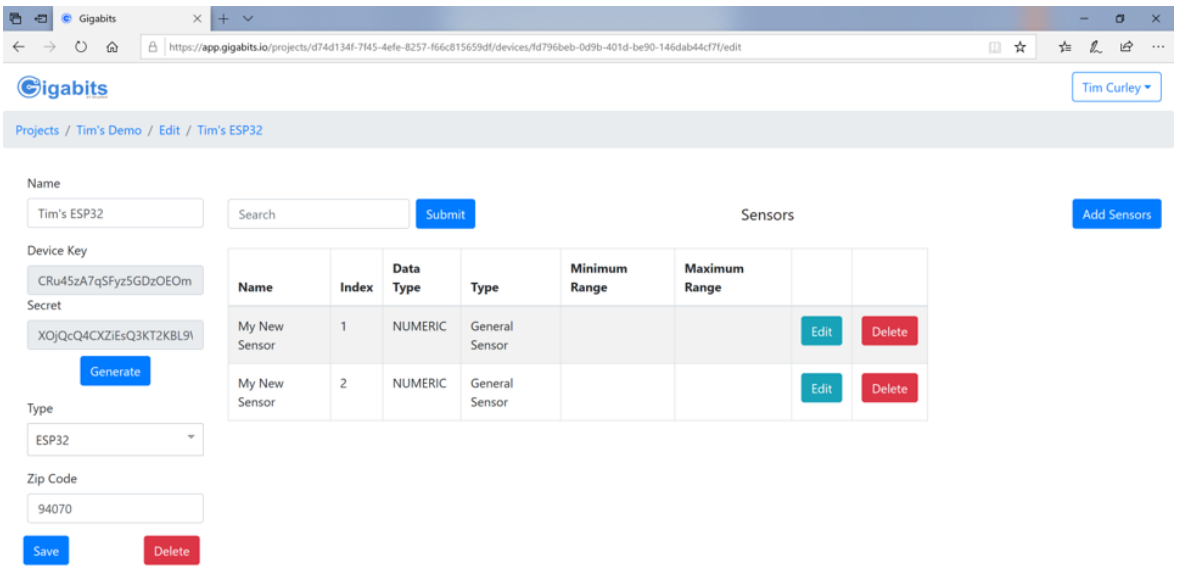
Software Figure 12

Clicking on the "Add" button will take you back to the page used to edit devices. The device will now have one sensor.



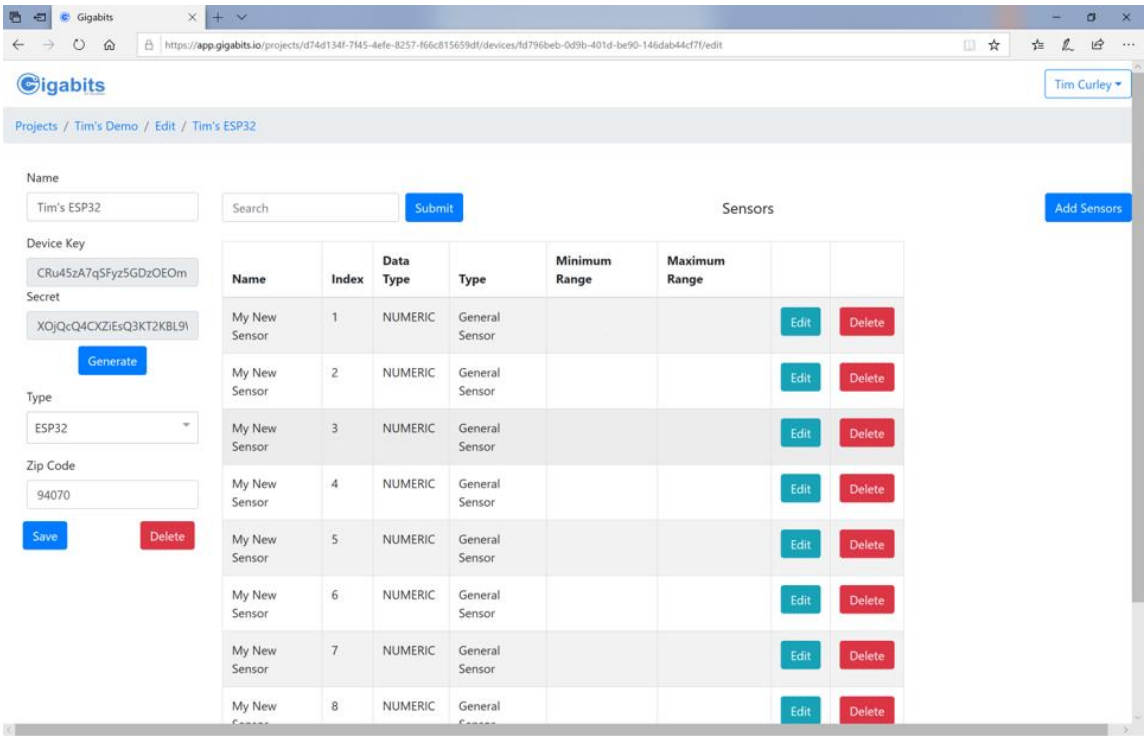
Software Figure 13

Software Step 14: Add another sensor to your device by clicking on the "Add Sensors" button again, then clicking on the "Add" button on the next page again.



Software Figure 14

Software Step 15: Repeat the "Add Sensors, Add" sequence 6 more times for a total of 8 sensors and one actuator.



Software Figure 15

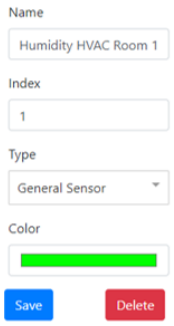
Software Step 16a: Go through all the sensors again. This time, give each sensor a useful name and a color. The name will have two parts. The first part describes the physical sensor type, like “HCPA-5BV-U3” or “TSL2651”. The second part is used to identify the sensor, like “HVAC, Room 1”. Each sensor type is statically assigned a number called the “sensorIndex” by the system. Each sensor value pushed to the Gigabits server is accompanied by a sensorIndex. That tells the server how to treat the value. Here is the current mapping from sensor type to sensorIndex. This will grow as sensors are added.

There’s currently a loophole that we take advantage of. As shown, each sensorIndex corresponds to a particular sensor type. The loophole is that the system doesn’t check the sensor type. That means that you can usually replace one sensor with another that has the same class of output (e.g. Numeric) and the system will dutifully display the new sensor’s output. You can use the sensor’s name to describe the actual sensor output.

Sensor Type	Sensor Index
HCPA-5V-U3, Humidity output	1
HCPA-5C-U3, Temperature output	2
Invert Display button (actuator, not sensor)	3
MPL115A2 Barometric Pressure	4
MQ2 Propane, Butane, Methane, Hydrogen, Alcohol Sensor	5
Soil Moisture Sensor	6
TMD26721 Proximity Sensor	7
TSL2561, Visible Light output	8
TSL2561, Infrared Light output	9

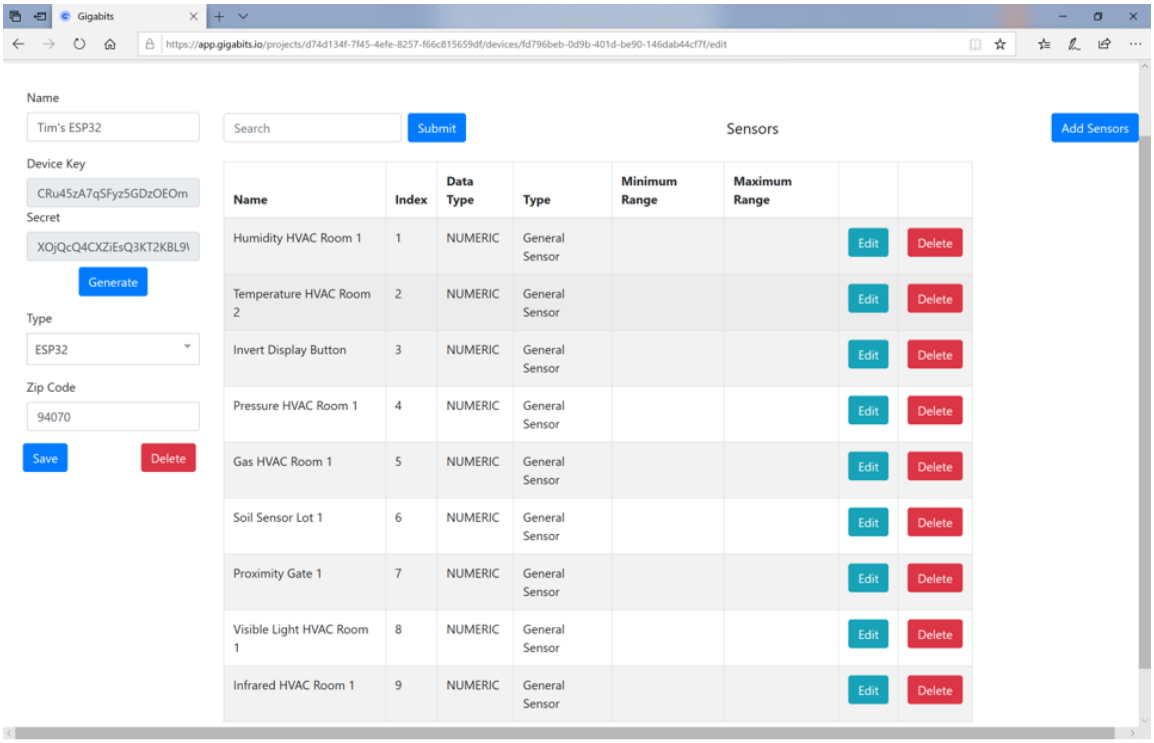
Table 1

Software Step 16b: While you're giving a sensor a name, you should also assign it a nice color. The color will be used to quickly identify gauges and chart lines. The mechanics of assigning a color to a sensor varies between web browsers. In all cases, you click on the color box, then use a browser-specific method to assign a color.



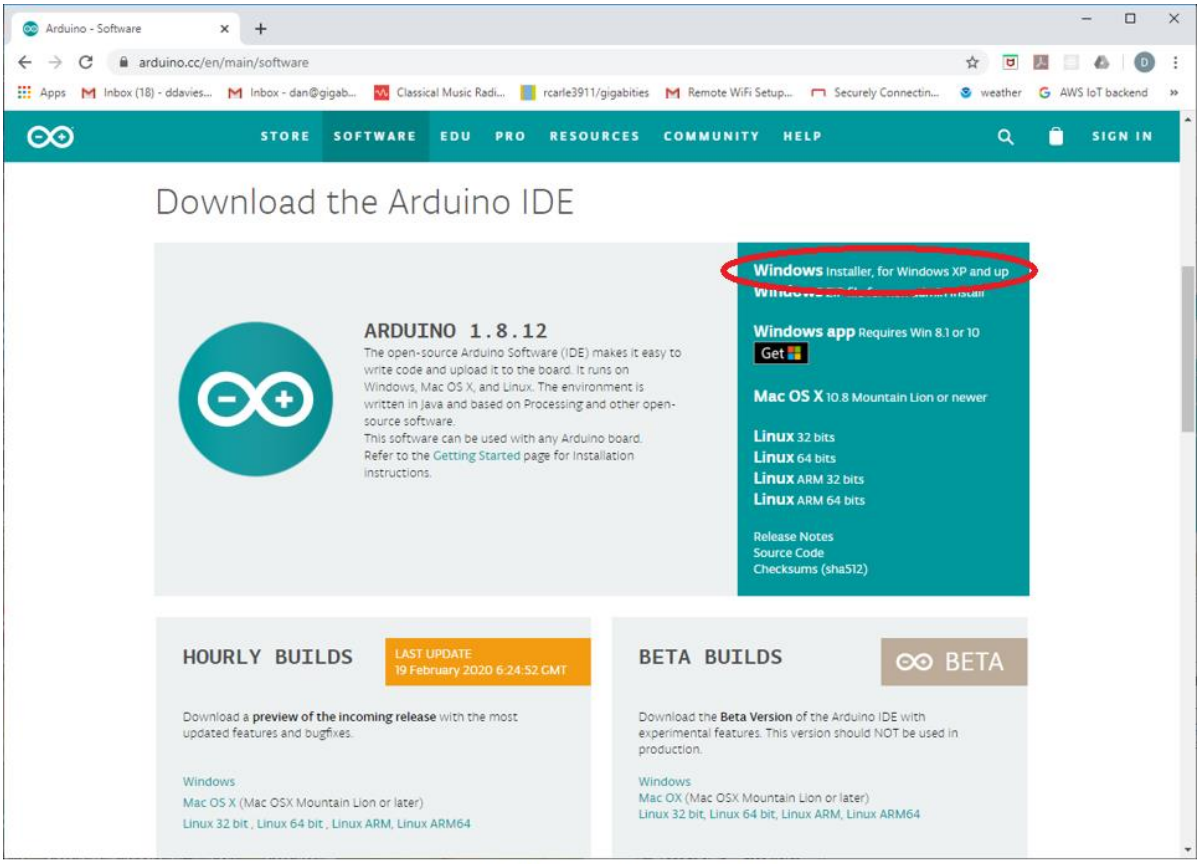
Software Figure 16

Software Step 16c: Here's a device with all its sensors. The Invert Display button isn't a sensor. It's a button that causes the little display on the Training board to change from normal video to reverse video and back again. It's our sample actuator.



Software Figure 17

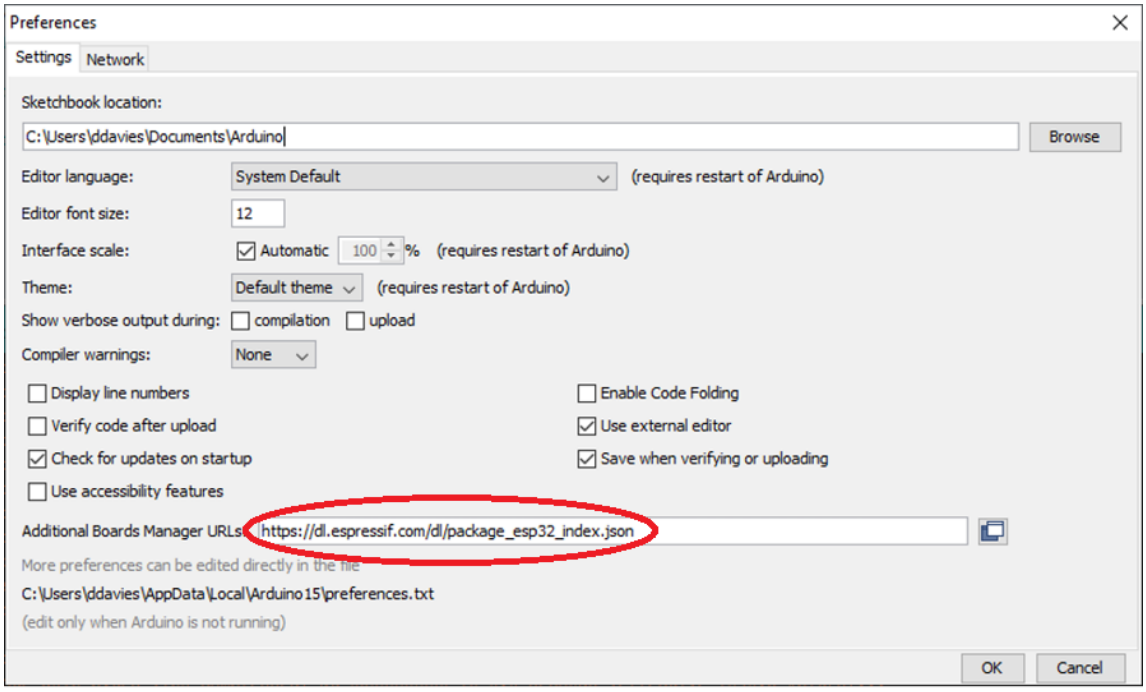
Software Step 17: Now that the dashboard is configured, we can start loading software. Start by loading the Arduino IDE. Open a web browser to https://www.arduino.cc/en/main/software.



Software Figure 18

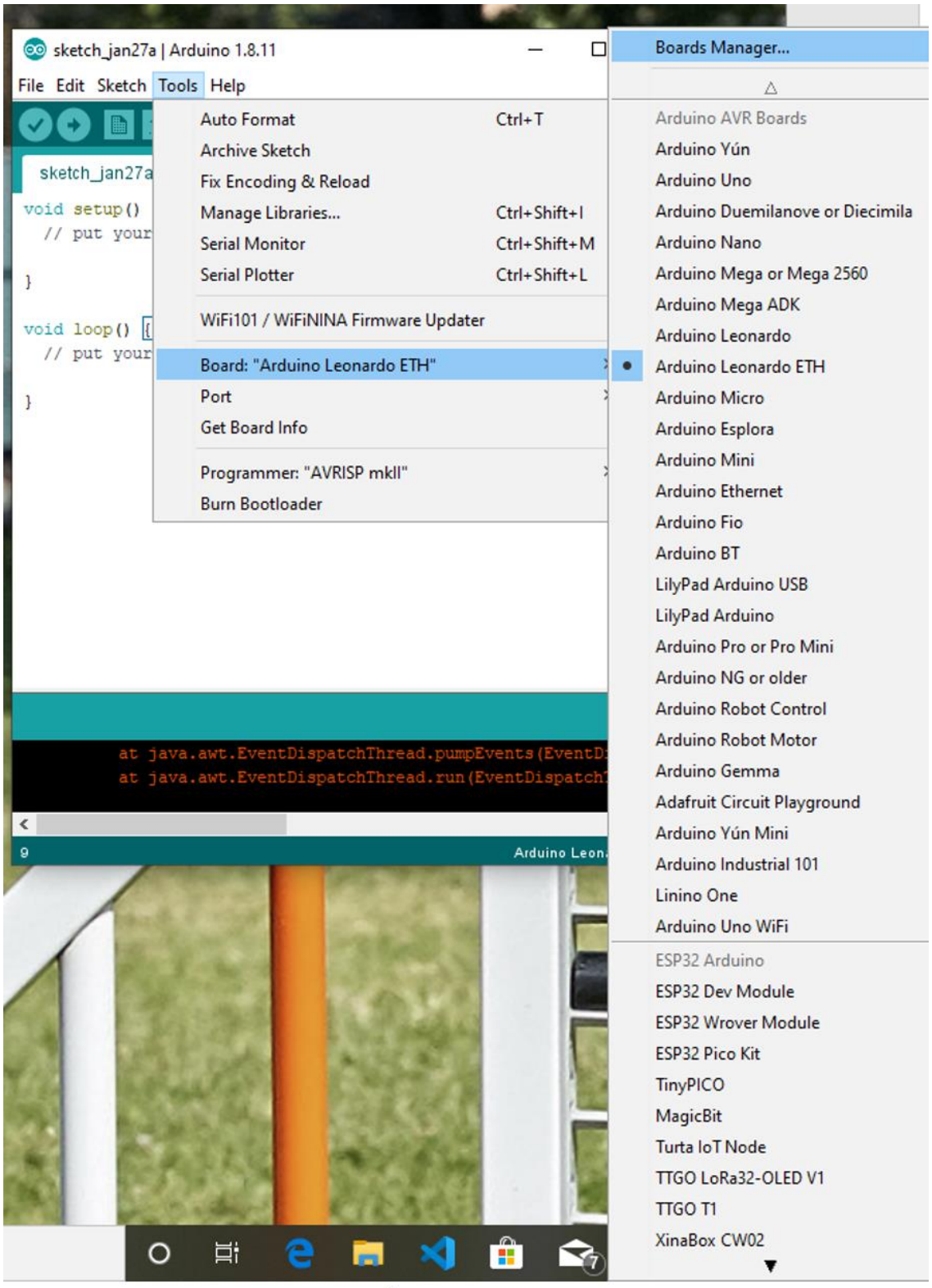
Using the script “Windows Installer, for Windows XP and up”, install a copy of the Arduino IDE.

Software Step 18: We need to load the code that links the ESP32 boards with the Arduino IDE. Open File -> Preferences. Put the string https://dl.espressif.com/dl/package_esp32_index.json into the text box labelled “Additional Boards Manager URLs:”. Close the Preferences window.



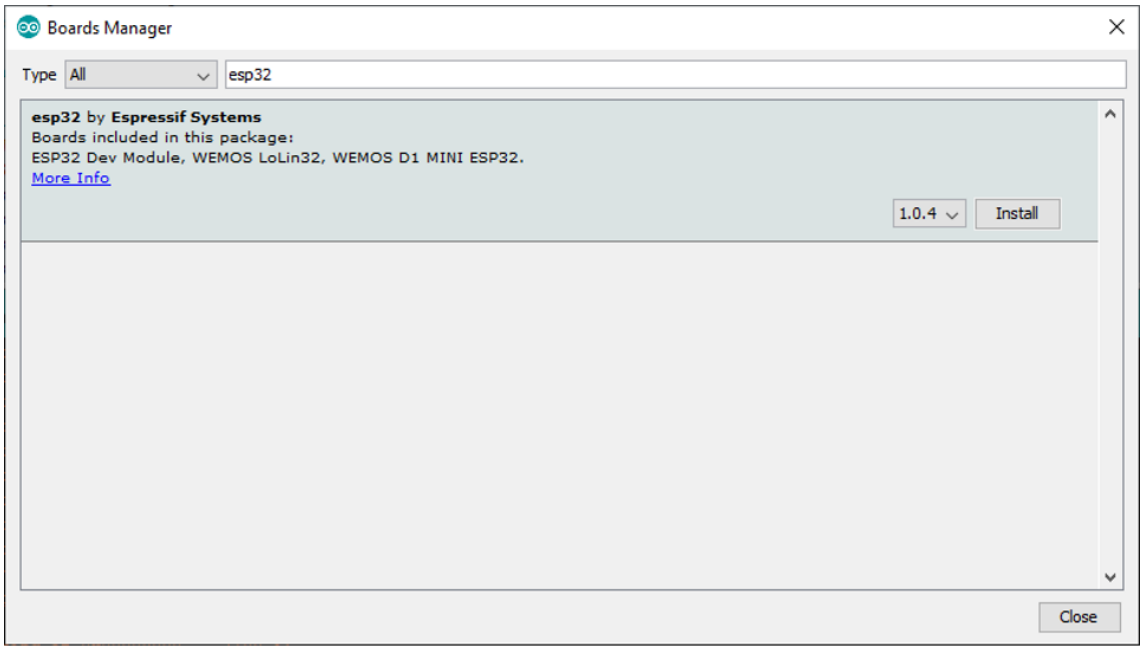
Software Figure 19

Software Step 19: Select Tools -> Board -> Boards Manager. Search for “esp32 by Espressif Systems”. Install this Board package. It will take a few minutes. It may ask if it can modify your PC. Say yes.

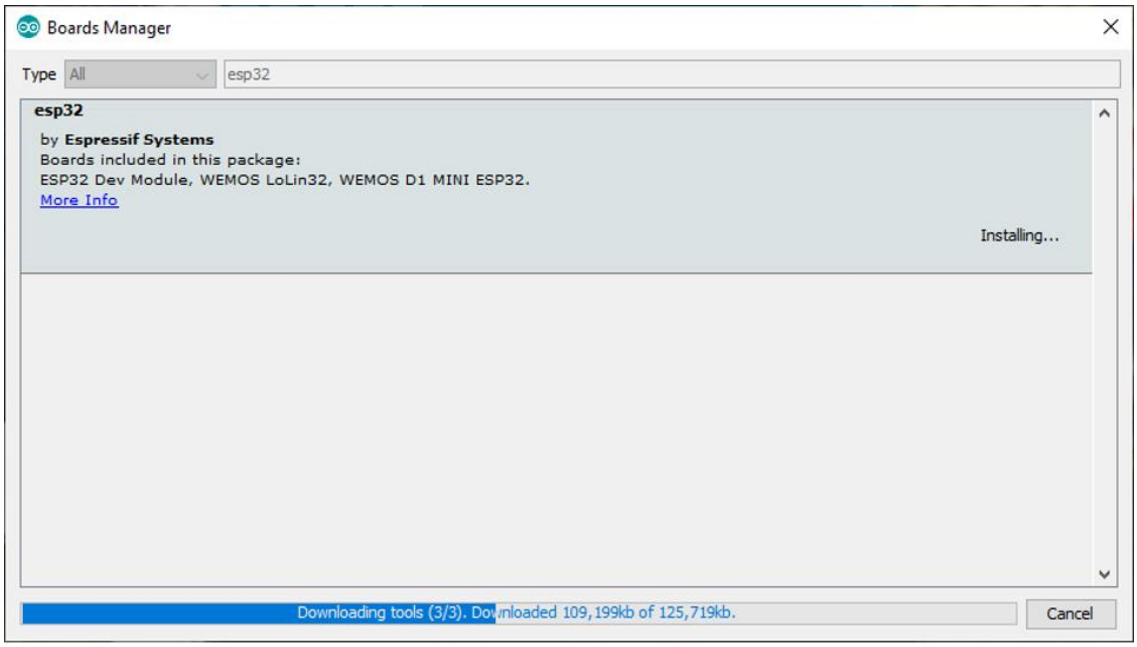


Software Figure 20

.

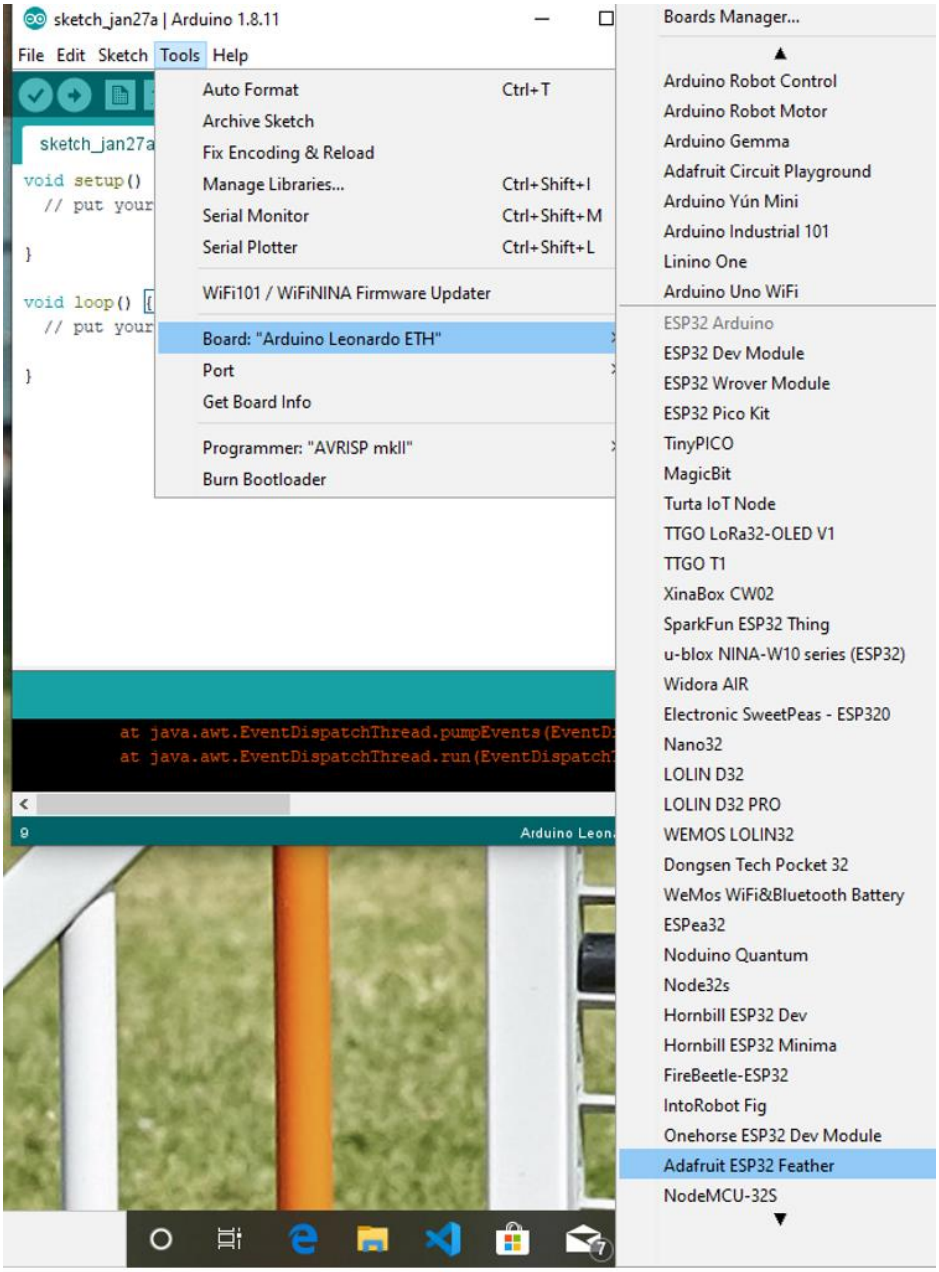


Software Figure 21



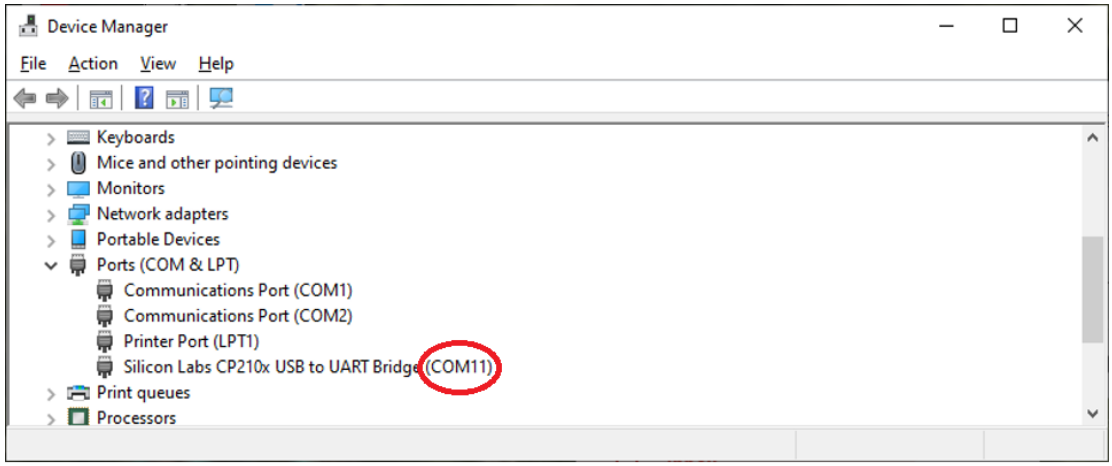
Software Figure 22

Software Step 20: The “esp32” package that you just installed includes many boards. In Tools -> Board, choose the board named “Adafruit ESP32 Feather”. In theory, it hardly matters what board you choose. In practice, we haven’t found another board that works with the ESP32 device and adaptor from NCD.



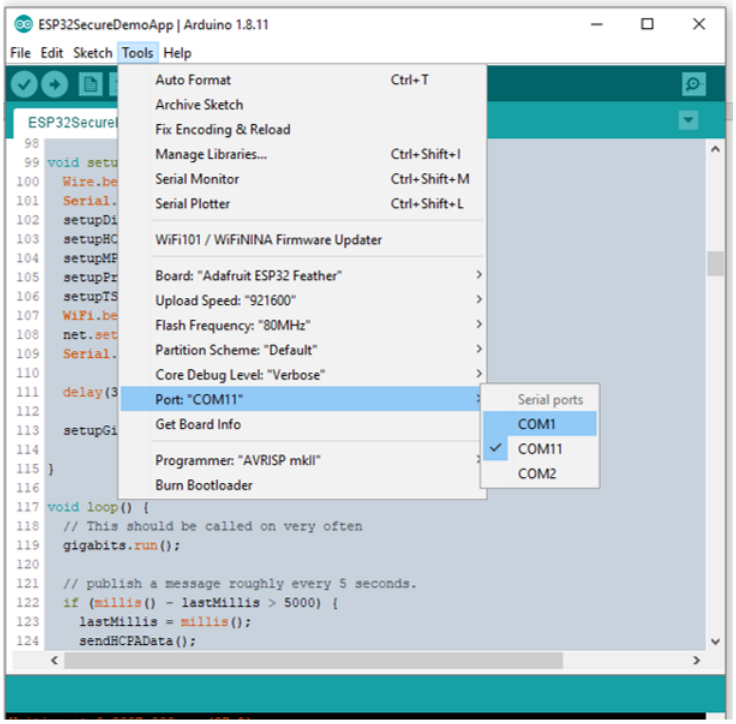
Software Figure 23

Software Step 21: Open the Windows Device Manager (On Windows 10, search for "Device Manager". Scroll down to "Ports (COM & LPT). There should be a port with an interesting name, like "Silicon Labs CP210x USB to UART Bridge (COM<a number>)". In this example, the number is 11. Any number other than 1 or 2 should work.



Software Figure24

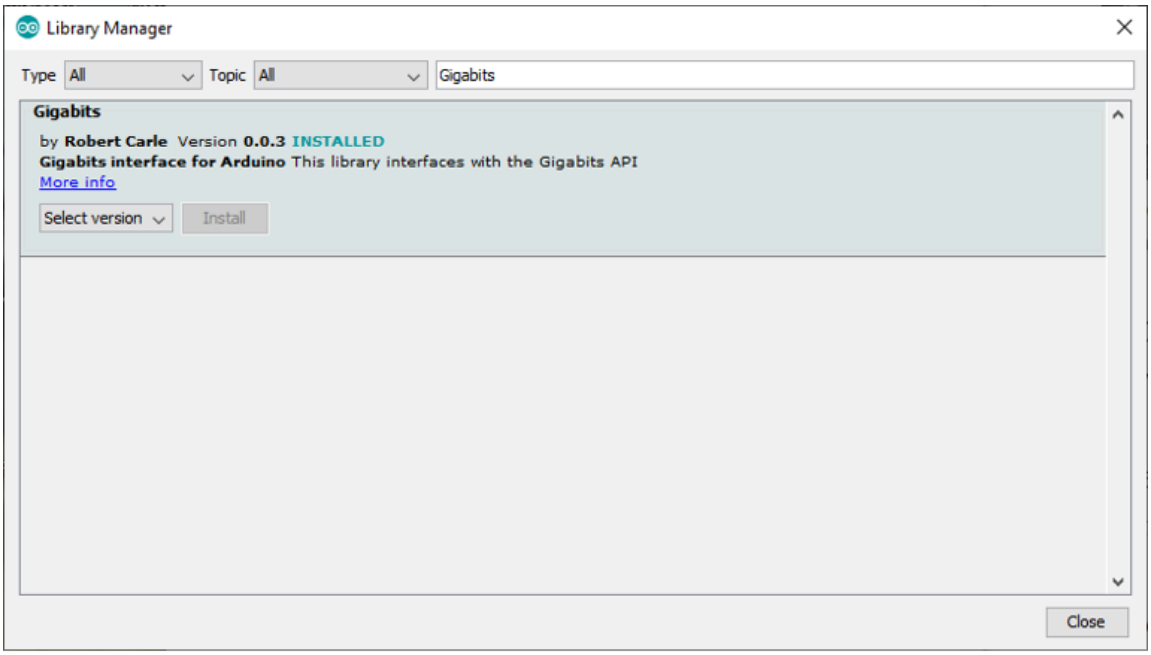
Software Step 22a: That's the Windows COM port that we'll use to communicate with our board. Back in the Arduino IDE, go to Tools -> Port and make sure the port shown is the port from the Device Manager.



Software Figure 25

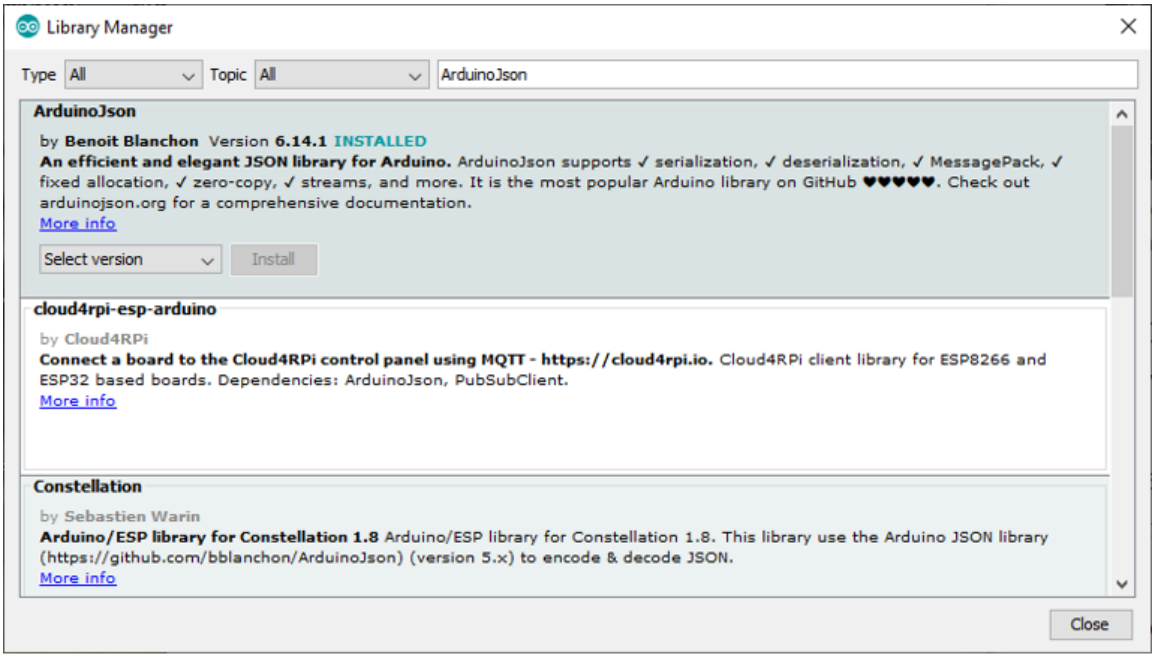
Software Step 22b: If not, press Tools -> Port and select the port from the list of ports. If you don't see your port's name, make sure the ESP32 is turned on and connected. If that's not the problem, try going into the Device Manager again, go down to Ports, right-click on the interesting one and select "Update Driver". If it's still not working, you might installing the drivers from https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers.

Software Step 23a: Use your Arduino IDE to fetch the "Gigabits" library (Sketch -> Include Library -> Library Manager; search for Gigabits). Press its "Install" button.



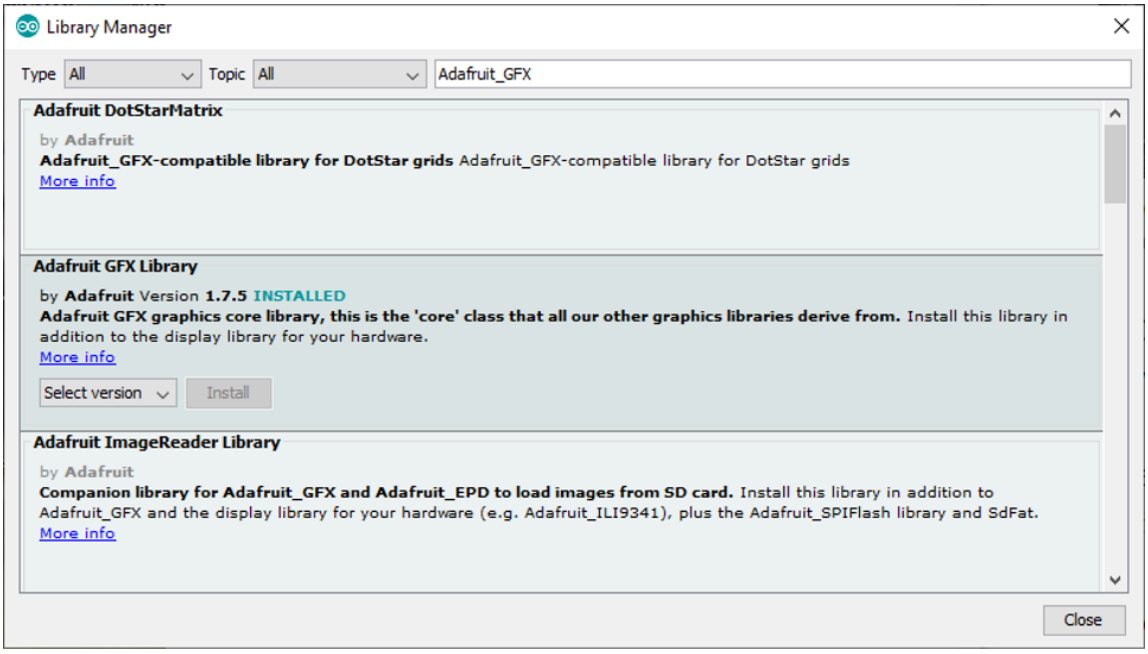
Software Figure 26

Software Step 23b: Install the "ArduinoJson" library.



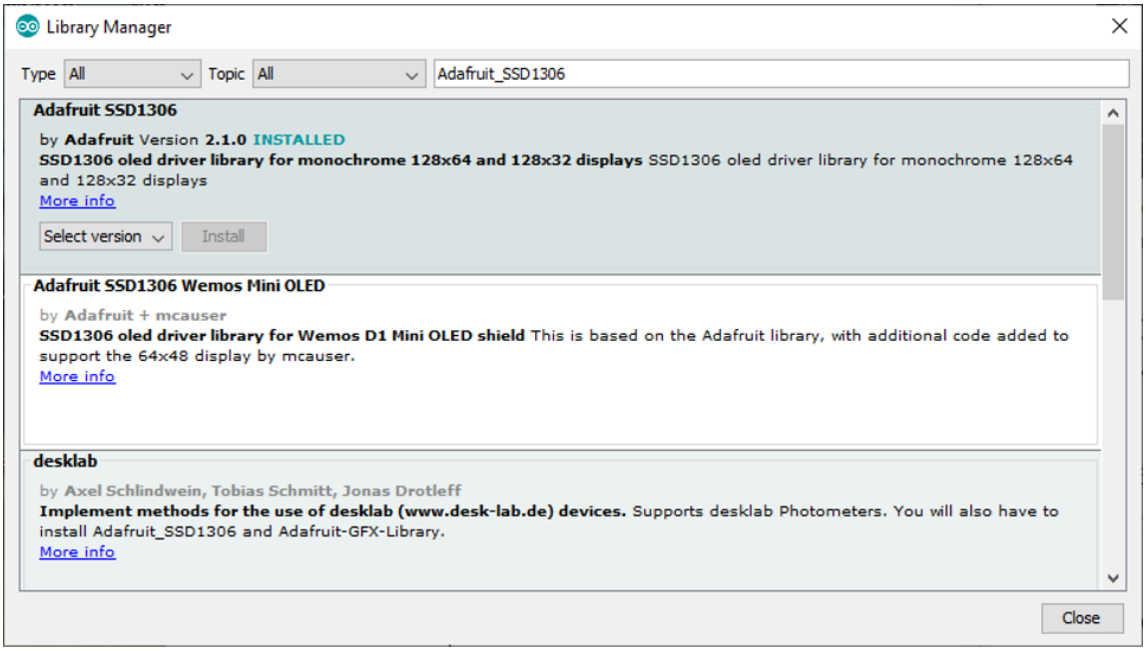
Software Figure 27

Software Step 23c: Install the "Adafruit_GFX" library.



Software Figure 28

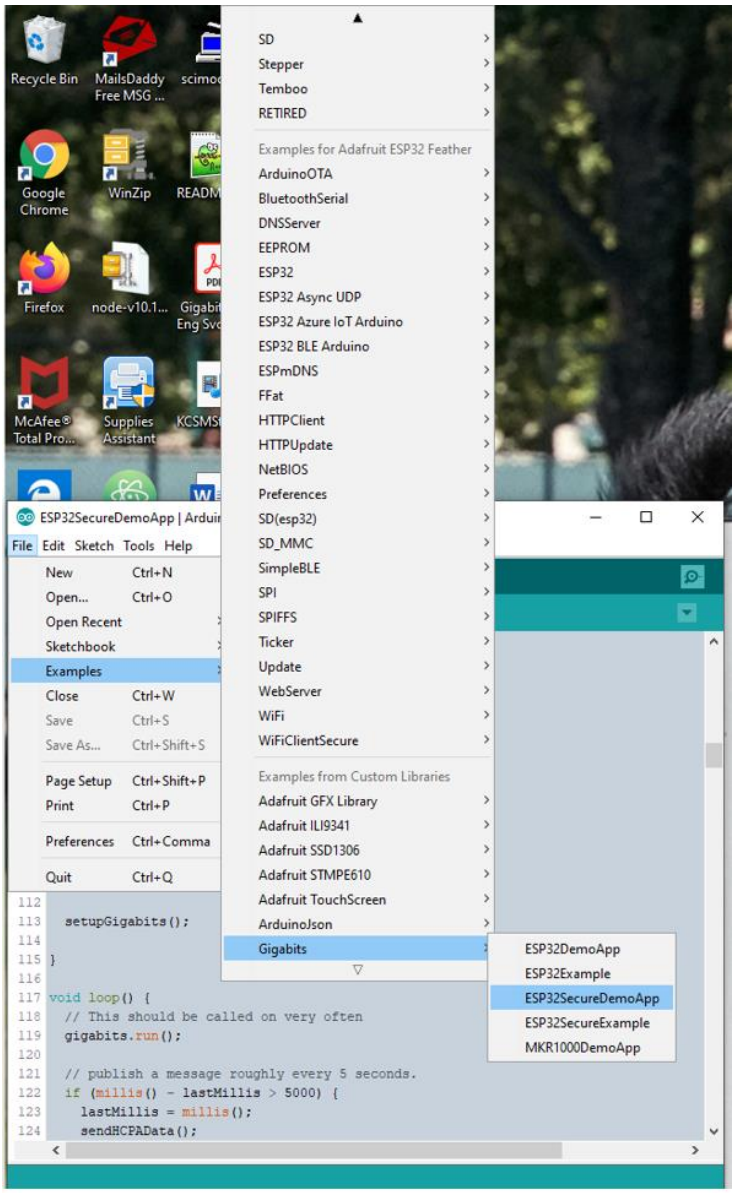
Software Step 23d: Install the "Adafruit_SSD1306" library. Close the Library Manager window.



Software Figure 29

Software Step 24: It's finally time to open the sketch (a program written in Arduino language). If you're writing your own sketch, you can use the rest of these instructions for inspiration. The important parts are including Gigabits.h, calling WiFi.begin with your router's ssid and password, calling gigabits.begin with your device's device key and secret (see below) and calling gigabits.sendRecord each time you want to send a sensor value to the server.

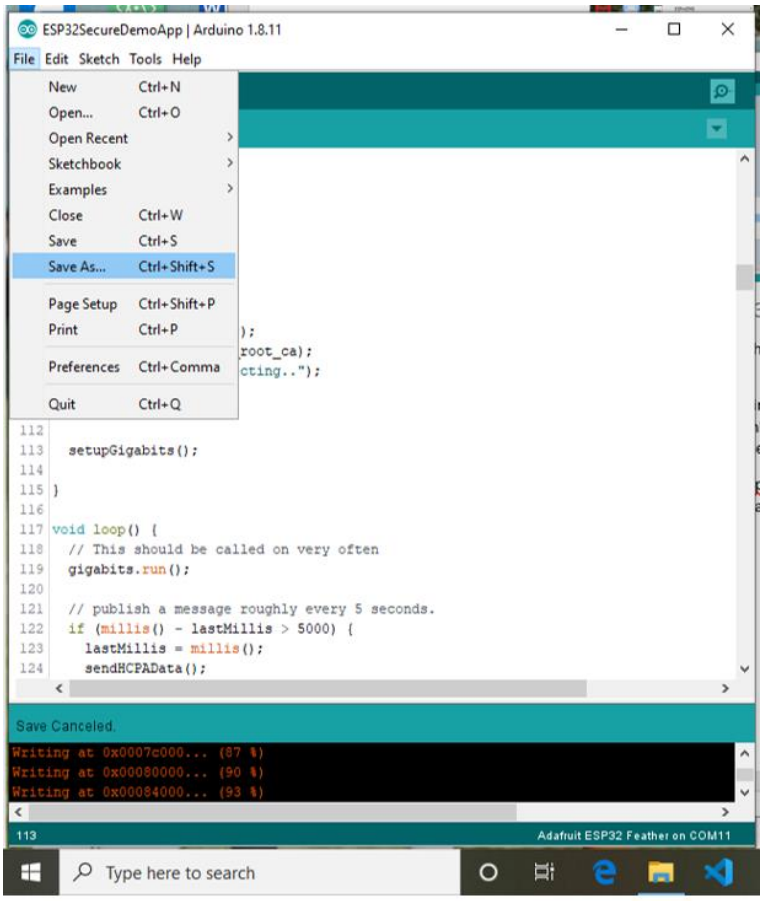
Software Step 25: Using the Arduino IDE, select File -> Examples -> Gigabits. Load the folder named "ESP32SecureDemoApp".



Software Figure 30

At this point, there will be two Arduino IDE windows open. Close the one that we didn't just open.

Software Step 26: We're going to modify the sketch inside that folder. We can't do that from where the folder is now because the directory is read-only. From the Arduino IDE, select "File -> Save As" to write the sketched elsewhere. Saving it on the user's desktop is convenient. That's at C:\Users\<your windows name>\Desktop



Software Figure 31

Software Step 27: In the version of ESPSecureDemoApp.ino that you just copied from the user's desktop, scroll down until you get to a line that says "Change these!"

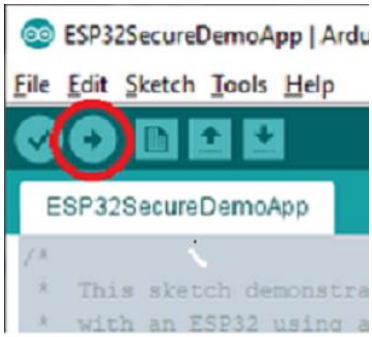
Software Figure 32

Change "ssid" to your router's name. Similarly, replace "pass" with your router's password.

Software Step 28: Way back up above, we declared a project that contained some devices. In the dashboard, go to Projects -> <your project's name> -> Edit. The project's "Device Key" and "Secret" are shown. Copy the Device Key for the project and insert it in place of "devkey" in our sketch. Similarly, replace the sketch's Secret with the project's secret.

Software Figure 33

Software Step 29: Finally, press the Compile and Upload button (in the upper left of the Arduino IDE window, it looks like an arrow facing to the right).



Software Figure 34

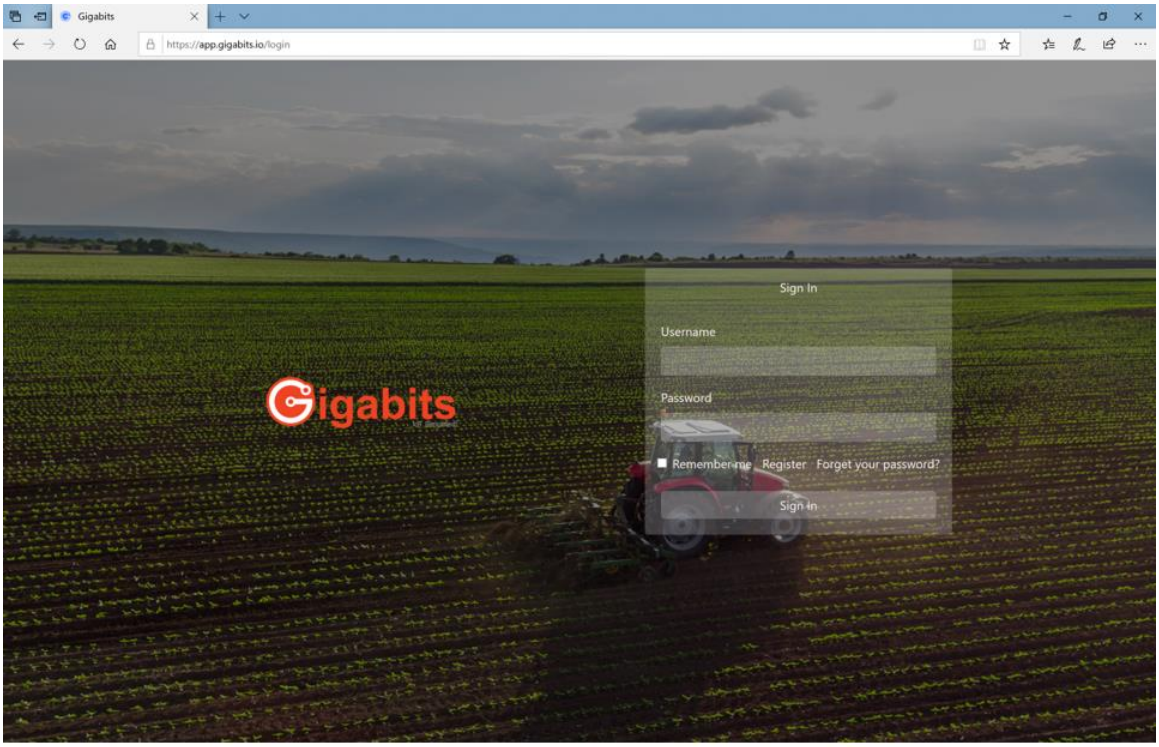
This will compile the sketch and start running it. It should start pumping sensor data to the dashboard. You should be able to see this on the Serial Monitor. You can open the Serial Monitor by selecting "Tools -> Serial Monitor"

You've completed the Gigabits setup and are ready to start configuring your dashboard's gauges and charts!

Set up your Dashboard

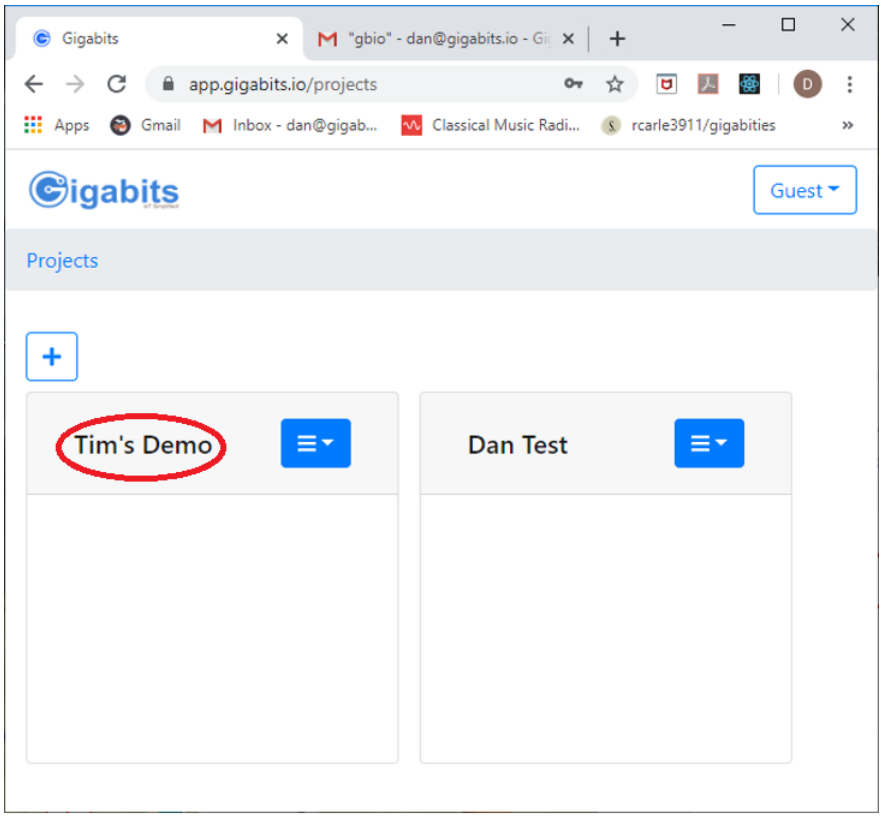
In this section, we will show how to set up charts, gauges and command buttons.

Dashboard Step 1: Open a web browser to “app.gigabits.io”



Dashboard Figure 1

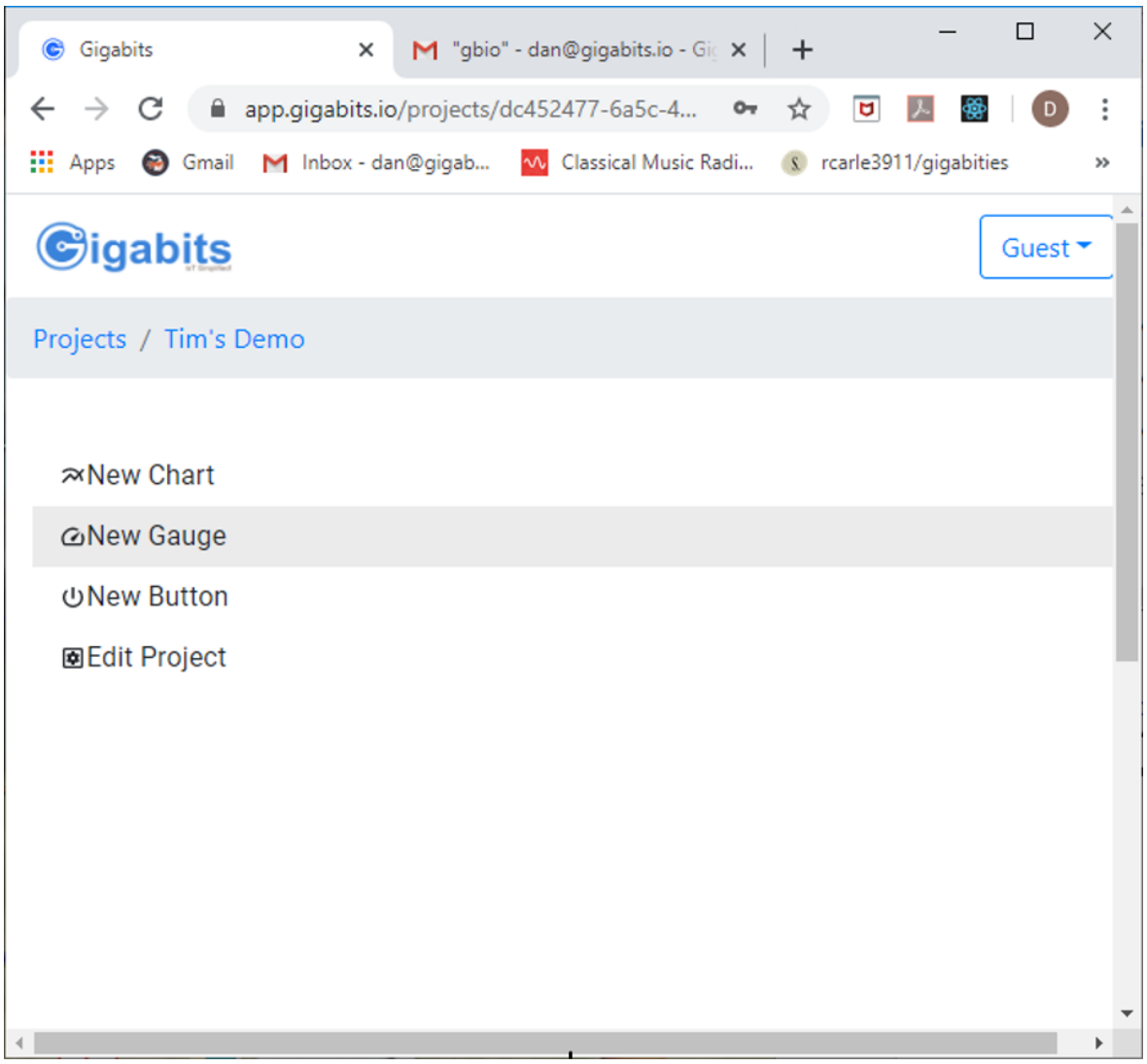
Sign in. That should take you to the Projects page.



Dashboard Figure 2

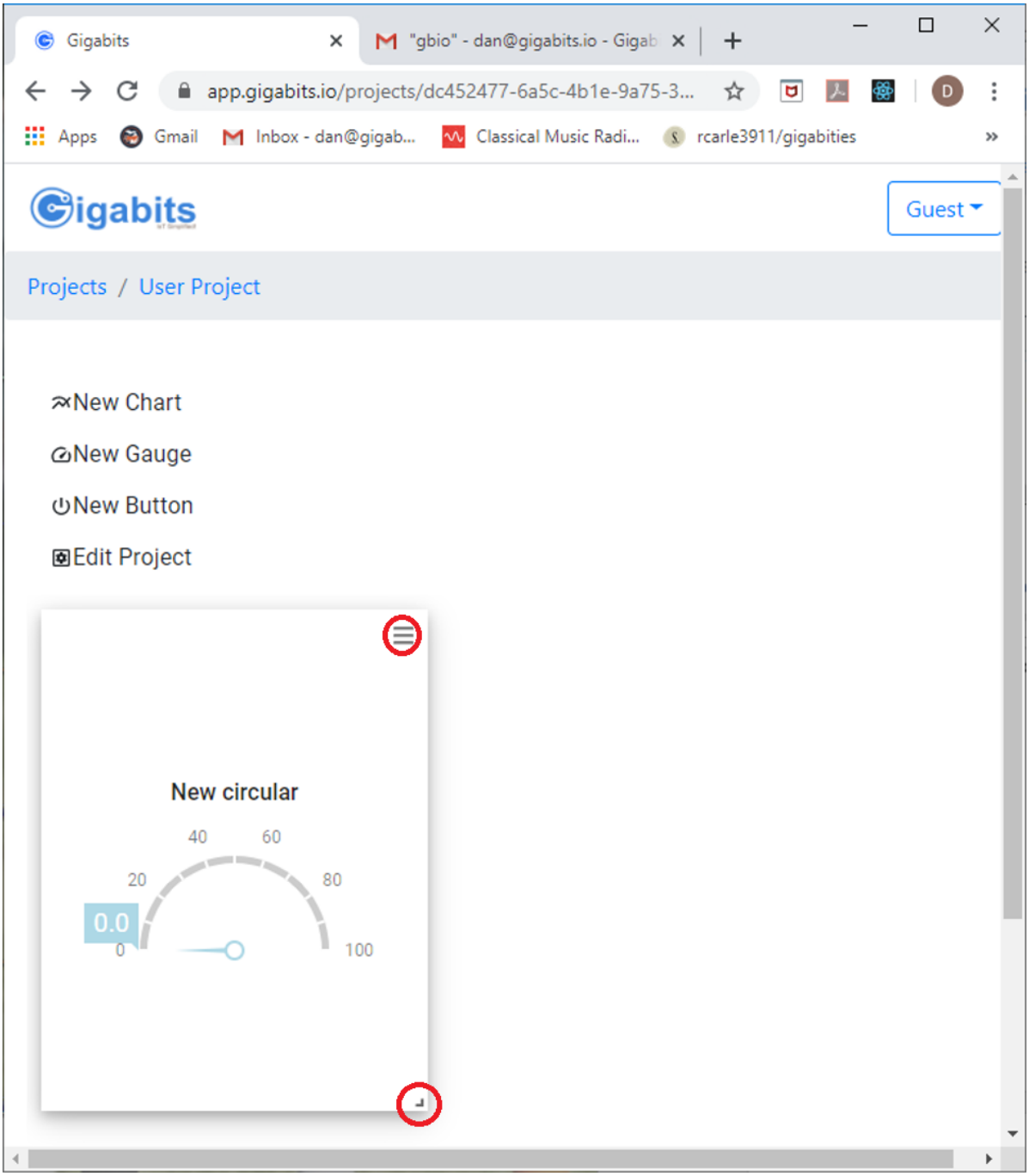
Dashboard Step 2: Click on the name of the project whose dashboard we'll set up. It's important to click on the project name, not the blue square next to that name. If you're following our example, click on the name "Tim's Demo", not the blue square next to "Tim's Demo".

Dashboard Step 3: On the left side of the User Project page, there are four commands New Chart, New Gauge, New Button and Edit Project. If the commands are chopped off, try making the browser window wider or narrower.



Dashboard Figure 3

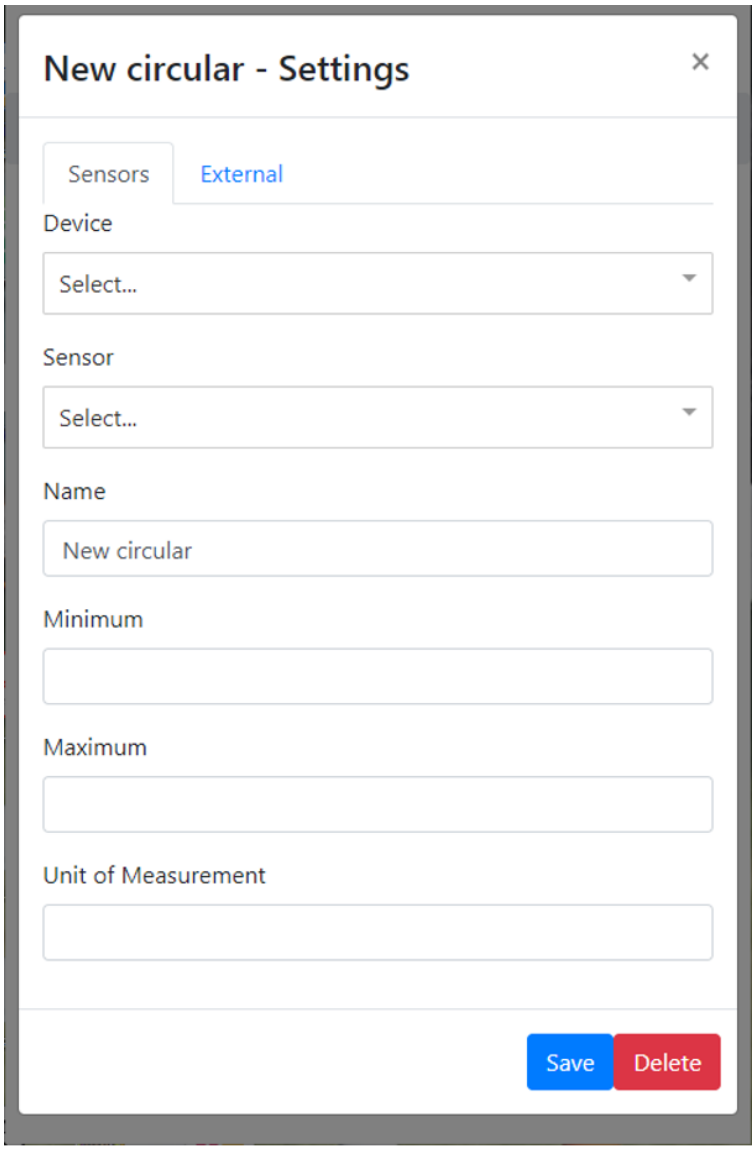
Dashboard Step 4: Let's start by making a new gauge. A gauge shows a sensor's current value. Click on the "New Gauge" command. A gauge will appear on the left side of the dashboard. It will be below the lowest gauge, chart or button, which can make it tricky to find because that location is often off the bottom of the screen.



Dashboard Figure 4

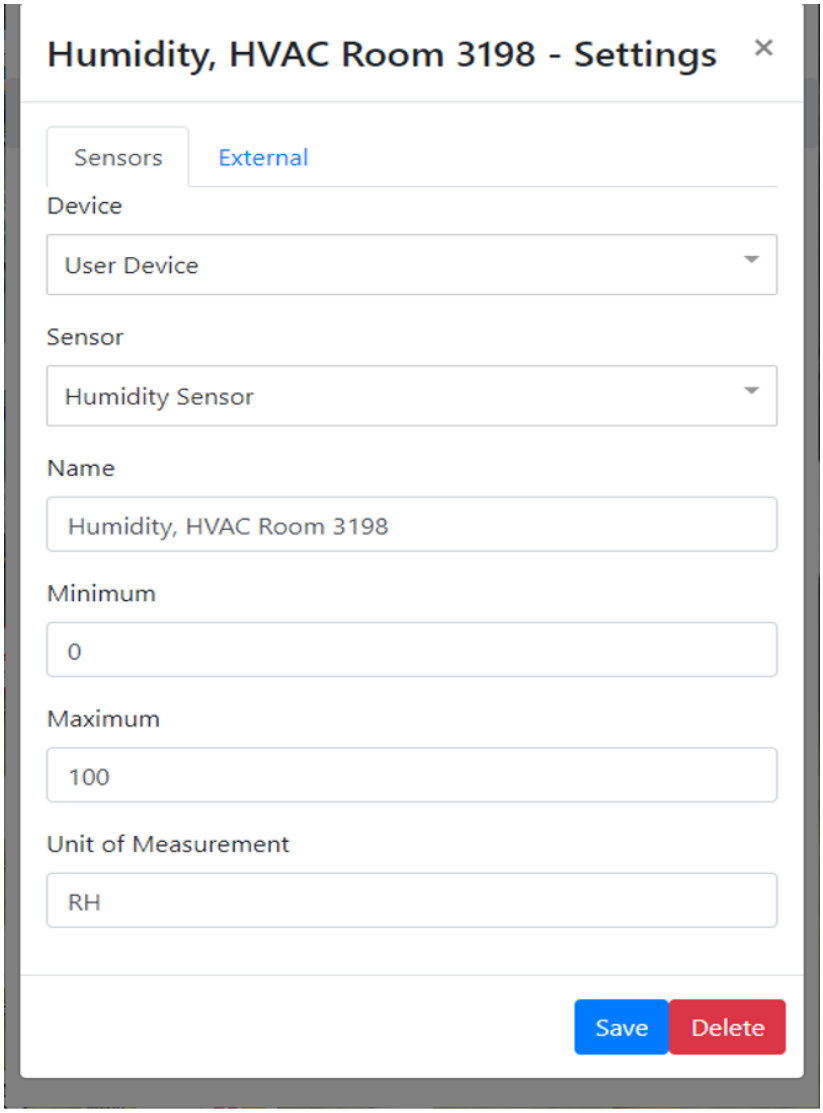
The gauge can be resized by dragging the tiny chevron in the bottom/right corner of the gauge. The gauge will have a configuration icon (three horizontal lines). Press the icon to configure the gauge.

Dashboard Step 5: The next page shows the settings for an unconfigured page. One of the first things to notice is that pressing the Delete button deletes the gauge.



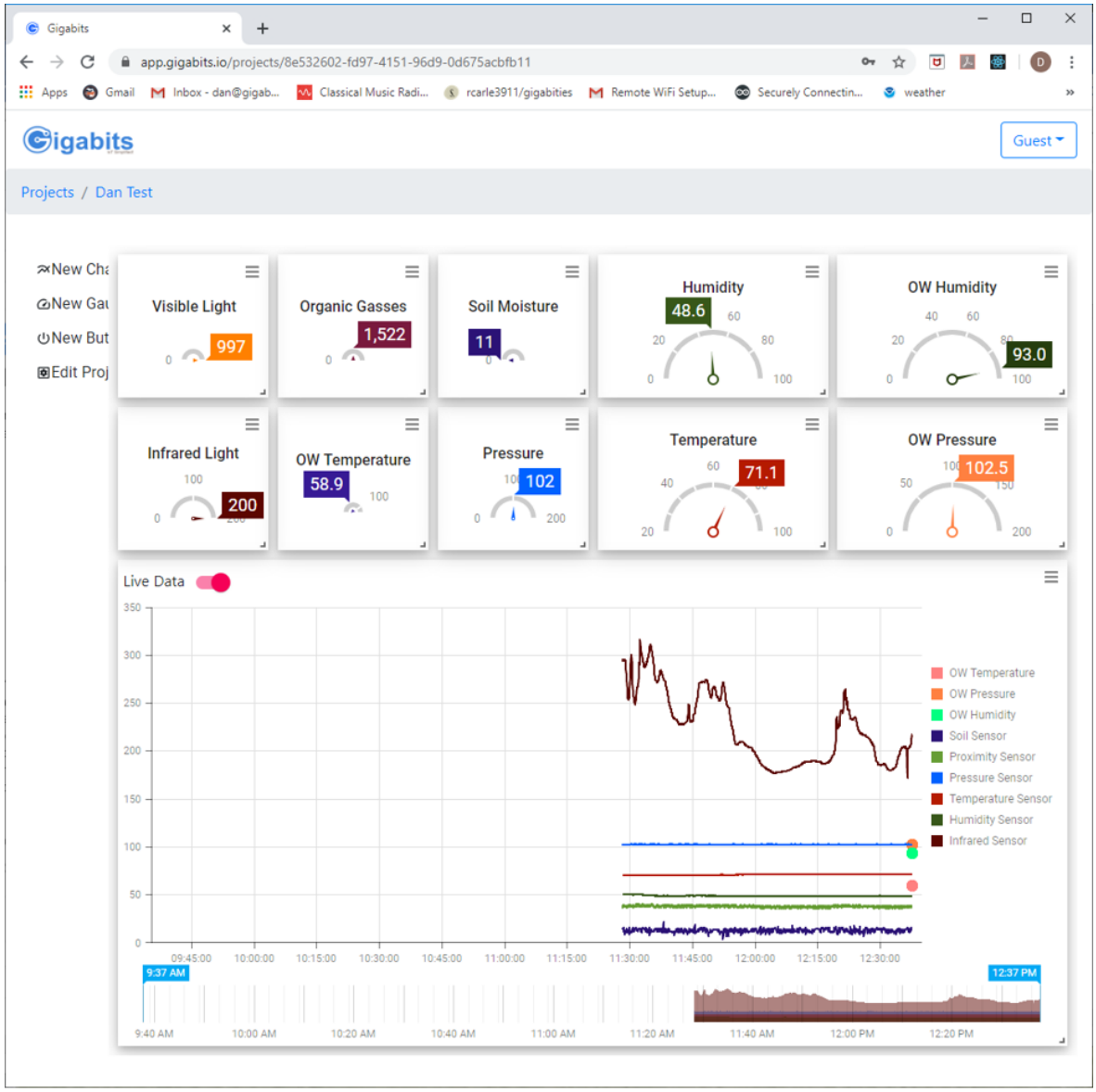
Dashboard Figure 5

Dashboard Step 6: Here's a configured gauge. It has the on-boarding information discussed above. The Device is from our project. The sensor senses humidity. The name is informative. The Minimum and Maximum bound the range of values shown on the gauge. The Unit of Measurement is used to label the chart axes. The color of the sensor data was chosen when the sensor was set up. Remember to press the Save button when you're done.



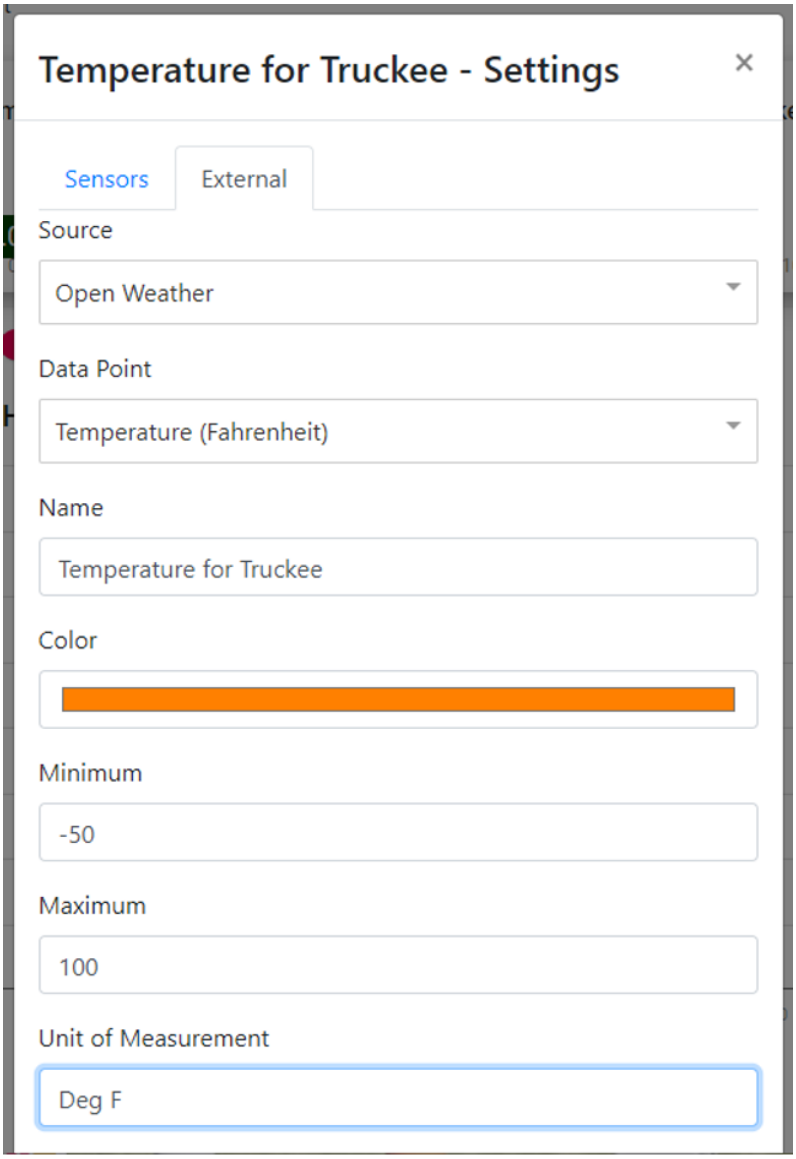
Dashboard Figure 6

Dashboard Step 7: To position the gauge on the dashboard, simply drag it around with the cursor. Dashboard items must be at the top of the dashboard or immediately under another item.



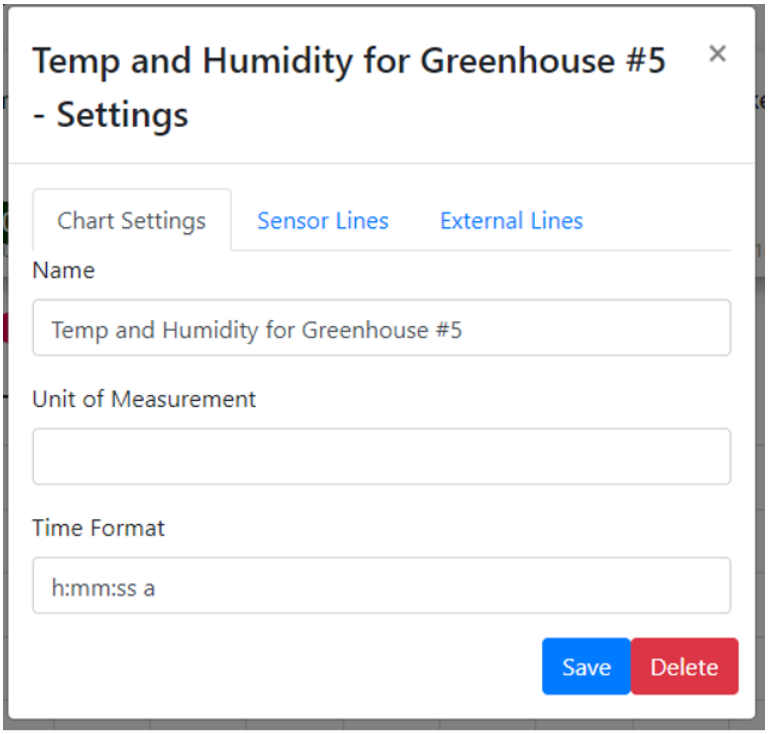
Dashboard Figure 7

Dashboard Step 8: You can also use gauges to display data from external databases. To illustrate this, Gigabits can display weather data. Go to your project and choose New Gauge. Press the new gauge's configuration icon. This time, click on the "External" tab at the top of Settings page. The only choice for "Source" is currently "Open Weather". There are 3 data types, Temperature, Humidity and Barometric Pressure. Pick an informative name. The weather data is for a US Zip Code. That Zip Code was chosen when the project was created. You can also select a nice color for the gauge, minimum and maximum values and a Unit of Measurement. Here is the Settings page for data from Open Weather.



Dashboard Figure 8

Dashboard Step 9: On to line charts! Press the "New Chart" command. An unconfigured chart will appear on the dashboard. Press its configuration icon. Give the chart a useful name. You can often leave the Unit of Measurement blank since the data being charted normally has different sources.



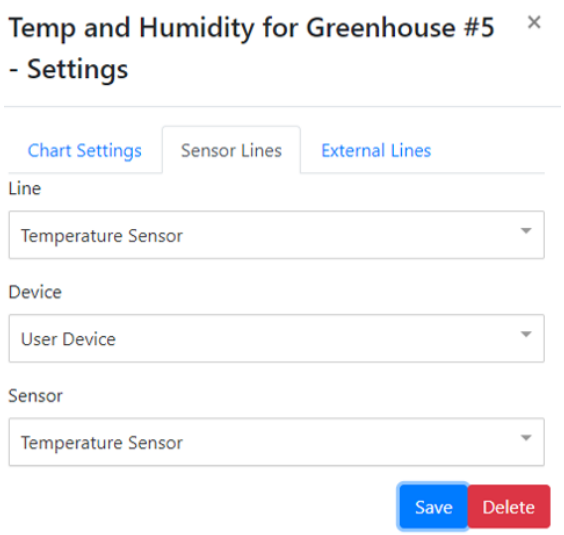
Dashboard Figure 9

Dashboard Step 10: You can either use a new chart line or modify an existing one. To begin, choose the "Sensor Lines" tab and set "Line" to "Create New Line". Fill in the Device and Sensor items as you did for gauges. Press the Save button. The window does not disappear because it's waiting for another chart line.

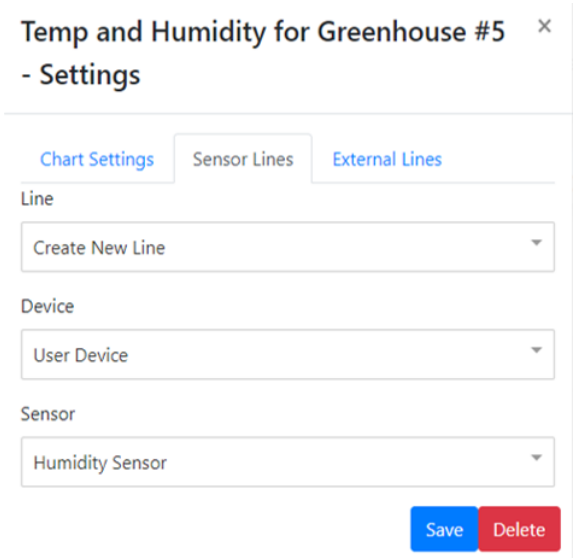
Dashboard Step 11: To close the Settings window, click on the "x" in its upper right corner.

Dashboard Step 12: To modify a chart line, open the Settings window, choose the desired line in "Line" and modify the remaining items.

Dashboard Step 13: To delete a chart line, open the chart's Settings, select the chart line in the Line menu and click the Delete button.

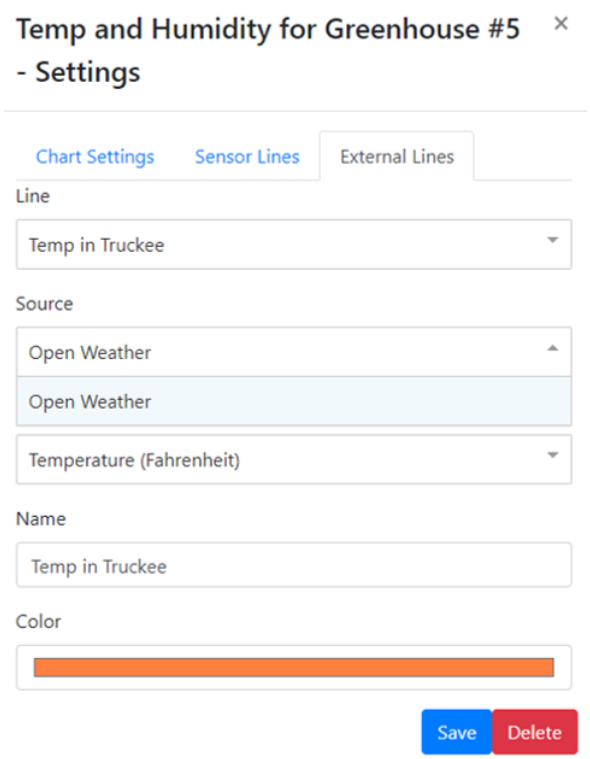


Dashboard Figure 10



Dashboard Figure 11

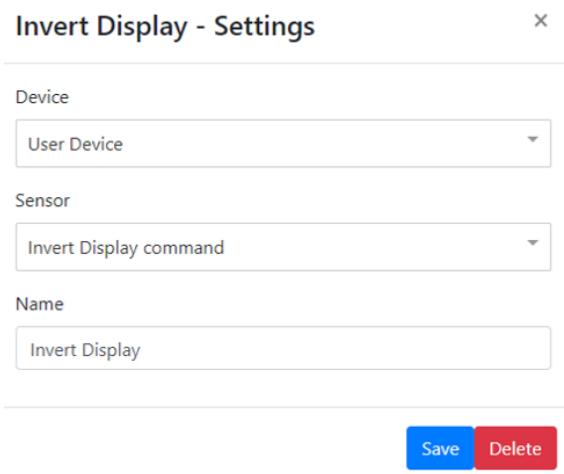
Dashboard Step 14: You can also create chart lines from external data. Press the "External Lines" tab on the Chart Settings page. Fill out "Line" as you did for chart sensors. Fill in Source and Data Point as you for the sensor gauges. Give the line a name and a color.



Dashboard Figure 12

Dashboard Step 15: And, finally, sending commands. In this demonstration, commands are used to demonstrate that Gigabits can control actuators as well as sensors. This actuator flips the display on the Training board between normal video and reverse video each time the button is pressed.

Dashboard Step 16: Press the New Button item on the dashboard. A new item will appear. Press its configuration icon. Fill in the current project's device. In the current implementation, actuators and sensors have some common code, so we sometimes call both "sensors." That's the case here. Since there is only one actuator (Invert Display command at sensorIndex 3) in the list of "sensors" for this project, the Sensor must be set to Invert Display command.



Dashboard Figure 13

Congratulations! You're ready to explore your data!