

# ARDUINO MKR NB 1500 ENVIRONMENTAL MONITORING GUIDE



# TELSTRADEV'S IOT STARTER KIT FOR ENVIRONMENTAL MONITORING.

Now you've got your hands on our IoT developer kit and environmental sensors from Arduino, let's put them to use! This guide will show you how to set up your Arduino MKR NB 1500 and ENV shield on Telstra's Cat-M1 IoT network and send sensor data to the Arduino IoT Cloud. At the end of this project you will be able to view the ambient Humidity, Light, Temperature and UV of your environment in real time all through a simple dashboard on desktop or mobile!

After completing this guide, you will have the tools to take on the hundreds of open source projects on Arduino's Project Hub or make a project of your own. If you are looking to scale your IoT solution for enterprise use, consider using [Telstra's Enterprise-Grade IoT Platform](#). We have [a guide](#) for integrating your Arduino with this platform.

The following sections all assume you've purchased an [Arduino MKR NB 1500 kit through our developer centre](#), have one of their Environmental Shields (sold separately) and have completed the TelstraDev Arduino MKR NB 1500 [Getting Started Guide](#) to set up your device and SIM.

## ENVIRONMENTAL MONITORING

CHAPTER 1	INTRODUCTION & SET-UP	3
	Cellular IoT	3
	Environmental Monitoring	3
	Set Up your Board & Connectivity	3
	Connect your ENV Sensor	3
CHAPTER 2	YOUR IOT DASHBOARD	5
	Add Device	5
	Add a Thing	6
	Add Properties	7
	Prepare the Sketch	9
	Create Dashboard	10
	Deploy and Monitor	11
	Develop and Improve	12
	Troubleshooting	12
	Summary	12

# Part 1

## INTRODUCTION & SET-UP

### CELLULAR IOT

The Internet of Things (IoT) is made up of billions of physical devices connected to the internet. Many home IoT devices connect over unlicensed networks like Wi-Fi, Bluetooth or LoRa. At Telstra, we provide connectivity via Low Power Wide Area mobile technologies like LTE-M (Cat-M1) and NB-IoT. Your Arduino MKR NB 1500 can connect to the internet via both awesome networks with a compatible SIM. It simplifies the need for configuring Wi-Fi networks, or being near a modem as Telstra's LPWAN networks have 4 million sq km of coverage in Australia! Cellular IoT is also more secure and reliable than unlicensed technologies. To learn more about IoT network comparisons, check out [these whitepapers](#).

### ENVIRONMENTAL MONITORING

IoT in Australia is revolutionising almost every industry. From farmers using temperature, location and water insights to manage their crops and livestock from paddock to plate, to postal services tracking the delivery status of their packages, and smart cities monitoring air quality and traffic, there are many things for us to gain from monitoring our environment.

The MKR ENV Shield included in Arduino's IoT Starter Kit includes sensors to monitor Humidity, Light, Temperature and UV. You could use these at home to monitor the conditions of your greenhouse or send alerts to turn on your light when the sun goes down.

### SET UP YOUR BOARD & CONNECTIVITY

Hardware needed:

- [Arduino MKR NB 1500](#) (from TelstraDev)
- Telstra Cat-M1 SIM (included with the board above from TelstraDev)
- RF Antenna (included with the board above from TelstraDev)
- [Arduino MKR ENV Shield](#) (sold separately)
- Micro USB cable
- Computer with [Arduino IDE](#) installed OR account with Arduino Cloud IDE

Take the Arduino MKR NB 1500 out of the box, and check you have a SIM and antenna.

Follow [TelstraDev's Get Started guide](#) for instructions on how to continue with the following steps:

1. Insert your SIM and install the antenna
2. Install Arduino IDE and libraries on your desktop
3. Set the correct radio access technology (Cat-M1)
4. Check the connectivity with example sketches

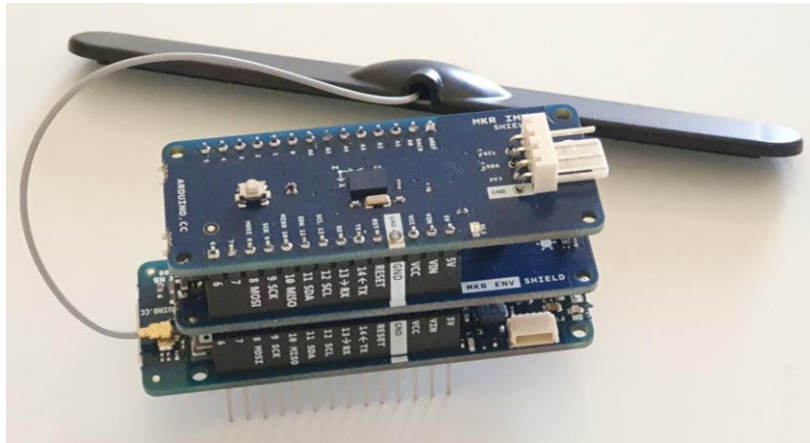
Now you have your device connected to the IoT network, let's connect some sensors to get some environmental data readings!

### CONNECT YOUR ENV SENSOR

The Arduino MKR series is built modular, so you can stack components on top of each other without breadboards and extra wires (save those for connecting LEDs and relays later!).

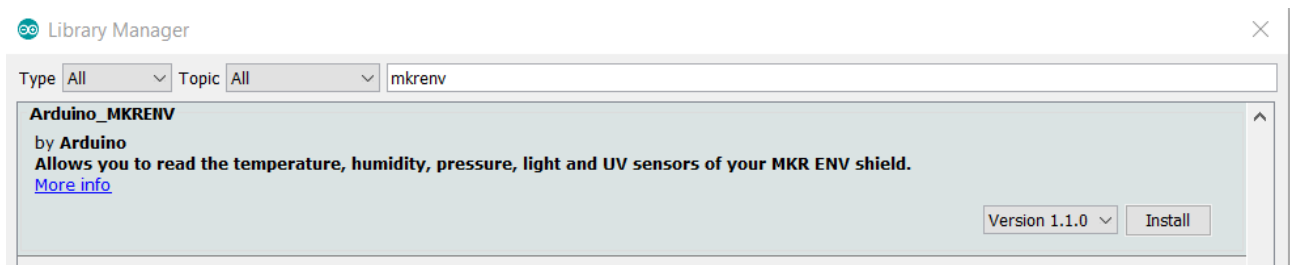
#### Connect the shield to the board

Align the MKR NB 1500 and ENV shield making sure that the labelled pin connectors are matching.

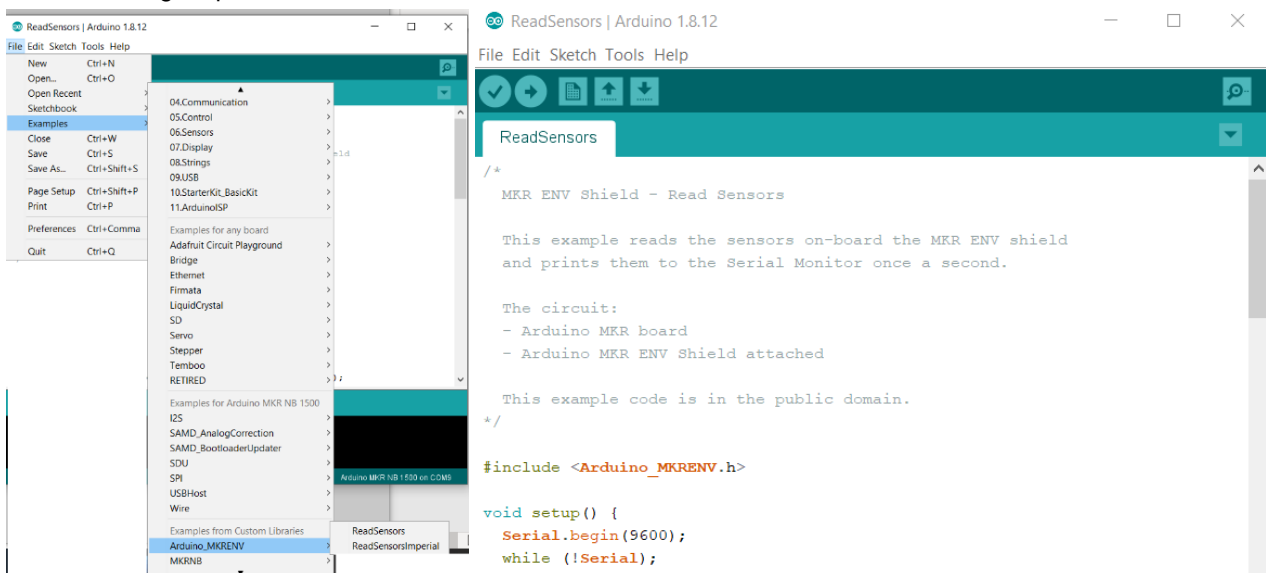


### Install the ENV library and run a test sketch (optional)

Just like you did for testing the MKR NB Connectivity in the [Getting Started Guide](#), find and install the Arduino\_MKRENV library.



In the Arduino IDE, go to File>Examples>Arduino\_MKRENV>ReadSensors. Check the serial monitor for the sensor reading outputs.



# Chapter 2

## YOUR IOT DASHBOARD

Now your device is connected to the IoT network, and recording environmental data correctly, it's time to display that data in a dashboard! Arduino has free trial of their simple and effective Create IoT cloud, which we will use to build our project: <https://create.arduino.cc/iot/>

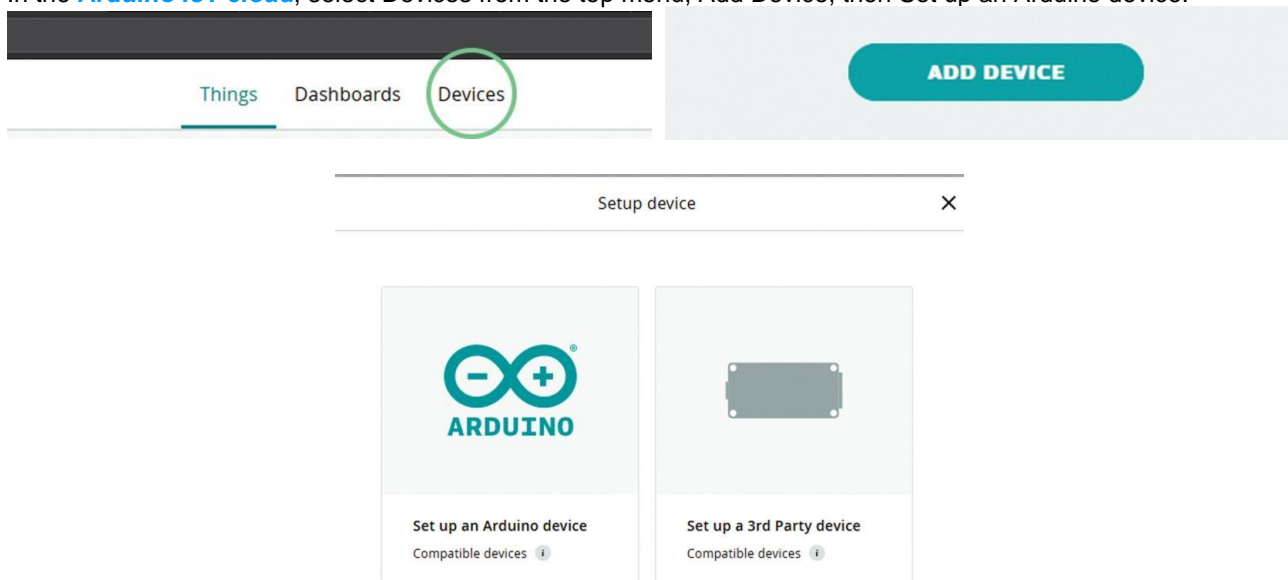
It's a simple process of Creating a Device, Creating a Thing, Creating a Sketch, and Creating a Dashboard. With these simple steps done, you will have a functioning IoT device on the Telstra M2M cellular network, which can send the sensor data from any location within [Telstra's 4 million sq. km of coverage area](#). An immensely impressive feat given the incredibly short amount of time it takes to get the program running.

After creating an [Arduino IoT cloud account](#), follow the Getting Started process to configure your dashboard and Arduino MKR NB 1500 board to communicate with the cloud. Having an account gives you the keys which will allow it to securely connect to our Cloud using an encrypted channel.

**Please note, the Arduino cloud website has been updated since these screenshots were taken and may look different to the below. But the steps are all the same!**

### ADD DEVICE

In the [Arduino IoT cloud](#), select Devices from the top menu, Add Device, then Set up an Arduino device.



Follow the prompts to install the Arduino Create Agent plugin. When the software has found a compatible Arduino device connected, select the configure button, and set a name for your device. This may take a few minutes to auto configure. Do not disconnect your device during this period. See our screenshots for the expected flow:

## Arduino MKR NB 1500 found

An Arduino MKR NB 1500 has been detected on port COM9 and ready to be configured.

CONFIGURE

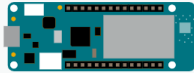
## Make your device IoT-ready

Your Arduino MKR NB 1500 will now be configured to communicate securely with the Arduino IoT. The process can take up to 5 minutes, please do not disconnect the device.



Uploading Sketch...

Setup device



Give your device a name


Name your device so you will be able to recognize it.

Device Name

TelstraDev\_IoT\_Board

NEXT

Setup device



Congratulations! You are all set

Your Arduino MKR NB 1500 TelstraDev\_IoT\_Board has been successfully set up. You can now connect your device to sensors, actuators and other inputs or outputs!

DONE

After a few minutes, if you have trouble with this step, disconnect and reconnect your board, make sure the Arduino Create software is running on your computer, and if that fails, check [this handy guide from DIYODE Magazine](#) to troubleshoot.

## ADD A THING

In Arduino Cloud, “Things” are the logical representation of a connected object. In this case, consider our Environmental Shield as the ‘Thing’, which has several properties like temperature and humidity. After creating your Device there will be a prompt to add a Thing, or else head to the top menu bar and create a new thing.

We will call ours MKR\_ENV\_Shield and associate it with the MKR NB 1500 device set-up in the previous step.

Note: in the free tier, only one Thing is supported per account. To add multiple Things (e.g. if you’d like to add a relay or IMU sensor) you’ll need to upgrade your Arduino IoT Cloud account.



## Create your first Thing

A Thing is a connected device that can communicate with the cloud. You can make your Things interact with other Things or anything else in the physical world.

[CREATE THING](#)

Setup

Sketch

Serial Monitor

### Variables

Variables are what you can monitor or control to make your Thing function. For example a temperature or a smart lamp. Once created, you can use them in your Sketch.

[ADD VARIABLE](#)

### Device

Select the device you want to use or configure a new one.

[Select Device](#)

## ADD VARIABLE

From the Thing setup tab, select the Add Variable button which will allow you to add temperature, humidity and the other sensor properties. This step automatically generates the Arduino code to manage the expected data from this variable.

In the pop-up screen, select the kind of variable you'd like to add and configure the settings. Some tips:

- Variable Name is the name that we need to reference in the code, so make sure it's easy to understand when you are looking over the code later.
- Most sensors will have a clear 'type' you can select to pre-load configuration. In this case we chose Temperature in Celsius.
- These are passive sensors that can't be written to (for example, an LED can be read/write), so these properties should be Read Only.
- When you set the preferences for updating sensor readings, keep in mind that when you send data from these sensors, it uses the remaining data balance on your IoT SIM (e.g. 10MB/month). If you can take sensor readings every 10s or 1 hour, rather than 1 second, you will use much less data.

Add variable

**Name**  
 Temperature

Temperature Sensor (°C) eg. 1 °C

**Declaration**  
 CloudTemperatureSensor temperature;

**Variable Permission**

☐ Read & Write
☒ Read Only

**Variable Update**

☒ On change
☐ Periodically

Threshold

1 °C

ADD VARIABLE

CANCEL

Repeat the process until all your sensors have been added. Note: for more than 5 properties, you'll need a paid Arduino plan.

MKR\_ENV\_Shield
⋮

Setup

Sketch

Serial Monitor

### Variables ADD

Name ↑	Last Value	Last Update
<input type="checkbox"/> Humidity <small>CloudRelativeHumidit...</small>	59.786	24 Oct 2020 14:40:14
<input type="checkbox"/> Lux <small>float lux;</small>	41.935	24 Oct 2020 14:58:23
<input type="checkbox"/> Temperature <small>float temperature;</small>	16.881	24 Oct 2020 14:43:37

### Device

**Odetta**  
ID: e4c9c8ea-2a3e-4d9d-8f16-...  
Type: Arduino MKR NB 1500  
Status:  Offline

Change

Detach



## PREPARE THE SKETCH

Once you are done, click Edit Sketch in the top right. You'll be redirected to the Arduino Online Editor with a pre-created sketch!

Note in the new version of Arduino cloud, to follow along with the tutorial you need to click 'Open Full Editor'.

The sketch will be named exactly like our Thing, plus the date of creation; e.g. `MKR_ENV_Shield_nov19a`. Other than the main .ino file, you'll see three more files:

- **ReadMe.adoc:** this is a plain text file with information about the sketch, the author, and the project itself.
- **thingProperties.h:** this is code generated by Arduino IoT Cloud when we added our properties. We don't need to edit this code, but it's useful to look at it as it shows which variables in our main sketch file (.ino) will be synchronized with the cloud.
- **Secret:** this tab allows us to fill in passwords that you may need for connecting to webhooks, APIs or Wi-Fi. Not necessary in this project.

For a further breakdown of each function and line in the pre-created code, refer to [this guide](#) for getting started with Arduino Cloud and a Wi-Fi IoT Kit (note, this MKR NB 1500 does not require any Wi-Fi configuration which makes things easier!)

There are a few things we need to personalise in the .ino sketch (e.g. `MKR_ENV_Shield_nov19a`), like initializing the Environmental Shield and storing the data from its sensors in your Thing variables. Make sure to check the variable names in the example code match your properties, as this simple sketch puts the desired sensor value into a variable, matching the variable name or names we created earlier.

If you are familiar with Arduino sketches, the rest is very simple. If not, you can refer to the sketch I've used, available on GitHub: <https://github.com/MichelleHowie/ENV-Monitoring-with-Telstra-and-Arduino-MKR/>

Because we have configured our device, thing, and properties, Arduino can handle the hard part!

### Introduce the MKR ENV Shield

With the default sketch, Arduino isn't aware of the MKR Environmental shield we have on top of the board (and any other things connected like relays, LEDs etc), so we need to include the ENV library. Find this in the libraries on the left navigation bar and include it. That should add this line at the start of the sketch:

```
#include <Arduino_MKRENV.h>
```



Now Initialise the ENV shield within the setup() function, by adding this code:

```
// Initialize ENV Shield
ENV.begin();
```

See our [GitHub](#) for the code sample you can copy. Your code should now look like this:

```
23 void setup() {
24   // Initialize serial and wait for port to open:
25   Serial.begin(9600);
26   // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
27   delay(1500);
28
29   // Defined in thingProperties.h
30   initProperties();
31
32   // Connect to Arduino IoT Cloud
33   ArduinoCloud.begin(ArduinoIoTPreferredConnection);
34
35   /*
36    * The following function allows you to obtain more information
37    * related to the state of network and IoT Cloud connection and errors
38    * the higher number the more granular information you'll get.
39    * The default is 0 (only errors).
40    * Maximum is 4
41    */
42   setDebugLogLevel(2);
43   ArduinoCloud.printDebugInfo();
44
45   // Initialize ENV Shield
46   ENV.begin();
47 }
48
```

## Refer to the sensors

In the main loop, add in the code that will update the sensor readings every `ArduinoCloud.update();`

```
void loop() {
  ArduinoCloud.update();
  temperature = ENV.readTemperature();
  humidity     = ENV.readHumidity();
  illuminance  = ENV.readIlluminance();
}
```

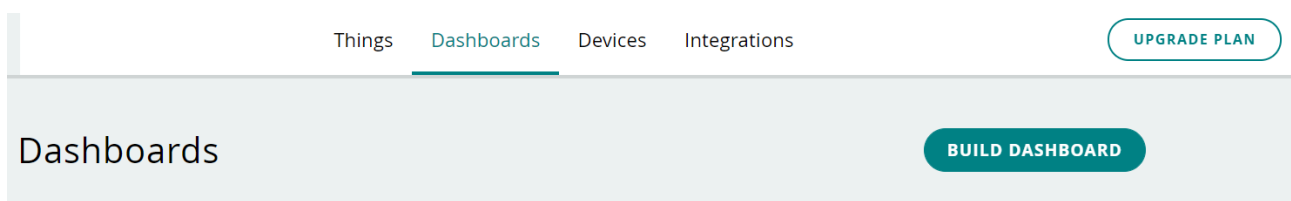
## Save, Verify, and Upload Sketch

Once you're happy with your code you can upload it to the development board. Plug the board into the USB on your computer and upload it directly from this web interface.

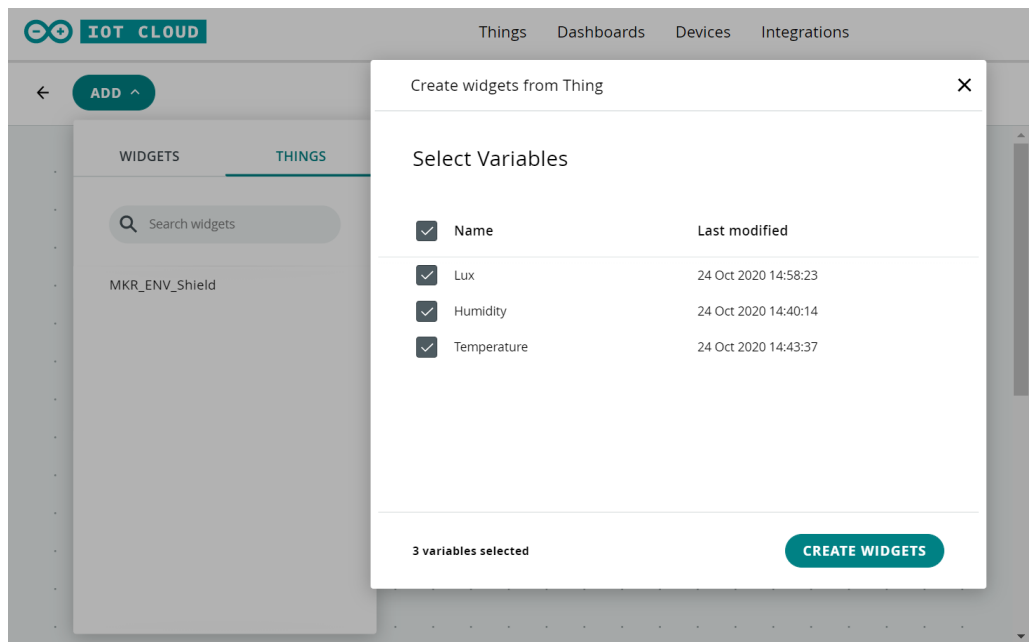
Your MKR Board is now sending data to the cloud! Let's create a dashboard to view it.

## CREATE DASHBOARD

From the Arduino Create IoT homepage, select the Dashboards button and select "Create Dashboard".



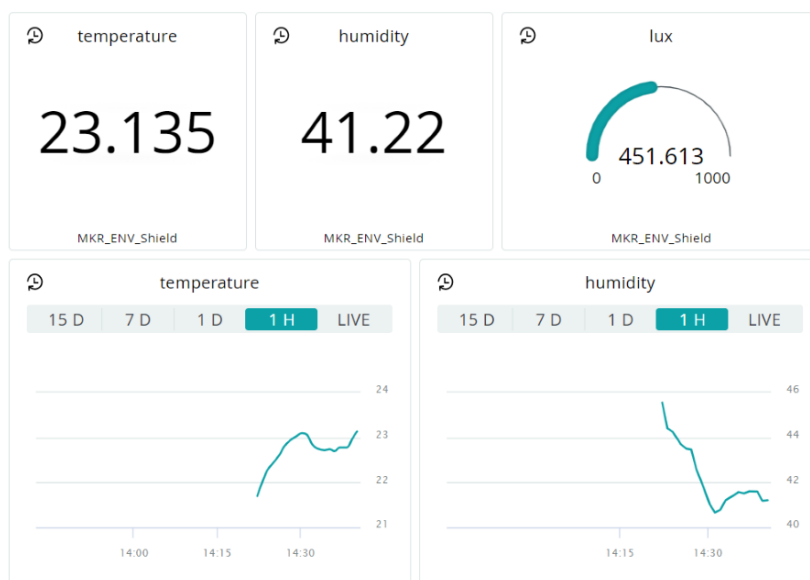
Add the variables from the thing you have created.



You're now able to add widgets that display the input and output of your connected devices. You could display the raw value of the sensor data (e.g. 24 degrees), or a gauge showing how high or low that value is compared to the min and max, or even a chart to show how the data changes over time. Get creative!

They are created with default values, so make sure you link the variable (property) in the right-hand side bar so it displays your live data!

#### ENV Shield



## DEPLOY AND MONITOR

One of the great features of the Narrowband IoT network, is that it is wireless! No need to bring your Wi-Fi modem, or ethernet cable around with your sensors. All you need is a compatible battery pack, connect that to the MKR NB 1500 Board, and off you go! We recommend looking into a weatherproofing enclosure or cover and stable batteries for the best results and to avoid damaging your board.

If you'd like to check your Telstra LTE-M/Cat-M1 coverage (there's 4 million sq. km!), check out our [coverage maps](#).

As you have loaded the sketch onto your board, even when you disconnect from the computer, it should continue sending sensor readings to the Arduino IoT Cloud! You can check back on your PC, or download the [Arduino IoT Cloud Remote app](#) on your smart device on the Apple Store and Google Play.

## DEVELOP AND IMPROVE

Now you have tested the sensors on your ENV shield, and you can monitor them from your dashboard or IoT Remote app, you can expand further and develop more complex projects with Arduino and Telstra IoT.

You could:

- Add in some code to check if your sensor reading is above a certain threshold, and send an alert
- Integrate other sensors to collect more data
- Add in relays or buttons to prompt actions from your board
- Actuate lights and buzzers in your IoT kit. Here's a handy guide: <https://create.arduino.cc/projecthub/133030/iot-cloud-getting-started-c93255>
- If you'd like to try the same thing with an analogue sensor, check out DIYODE's tutorial with a Gas Sensor: [https://diyodemag.com/reviews/long\\_range\\_iot\\_using\\_arduino\\_mkr\\_nb\\_1500\\_telsta\\_developer](https://diyodemag.com/reviews/long_range_iot_using_arduino_mkr_nb_1500_telsta_developer)
- Use temperature and humidity to predict a rainbow. Code Club did this with a raspberry pi emulator, but you can do the same logic with your Arduino: <https://medium.com/code-club-australia/getting-your-kids-coding-iot-523487c8253f>
- Set up polling via SMS, so you can ask your device for the live sensor readings! We've done this before: [https://github.com/MichelleHowie/IoTSensorData\\_OnDemand](https://github.com/MichelleHowie/IoTSensorData_OnDemand)
- Check out the Arduino Project hub for ideas from other makers around the world and share your own projects! <https://create.arduino.cc/projecthub/>

We at TelstraDev would love to hear how you go, share your project on our community forum: <https://dev.telstra.com/forums/iot-devices-forum>

## TROUBLESHOOTING

At any stage in your development, if you get stuck or come across some errors, check out our online resources:

- Review the [Frequently Asked Questions](#)
- Read the posts on our [TelstraDev forum](#) and the [TelstraDev blog](#)
- [Join the community](#) with our good friends at Arduino

If you're REALLY stuck, post your problem on our [TelstraDev forum](#), or [send us an email](#).

***Please note, the Arduino cloud website has been updated since these screenshots were taken and may look different to the below. But the steps are all the same!***

If

## SUMMARY

We saw in this guide how to set up your Arduino MKR NB 1500 and ENV shield on Telstra's Cat-M1 IoT network and send sensor data to the Arduino IoT Cloud. You will be able to view the ambient Humidity, Light, Temperature and UV of your environment in real time all through a simple dashboard on desktop or mobile!

Now it's up to you to take on your own project! Try some new sensors, actuators or IoT platforms. If you are looking to scale your IoT solution for enterprise use, consider using [Telstra's Enterprise-Grade IoT Platform](#).

We want to share your projects, no matter how big or small, to continue inspiring developers in the IoT community. You could sit alongside our case studies, how to's, guides and project ideas, on the [TelstraDev Developer Centre](#). Let us know what you're working on by emailing: [telstradev@team.telstra.com](mailto:telstradev@team.telstra.com)

## ACKNOWLEDGEMENTS

A few public resources made this guide possible:

- [DIYODE Magazine Nov 2020 article](#) on the Arduino MKR NB 1500 and guide to set up the Arduino IoT Cloud with Gas Sensor
- [Arduino Project Hub guide for connecting the IoT kit to Arduino Cloud](#)
- [Arduino Project Hub guide for getting Environmental Data on Arduino IoT Cloud](#)