blues wireless

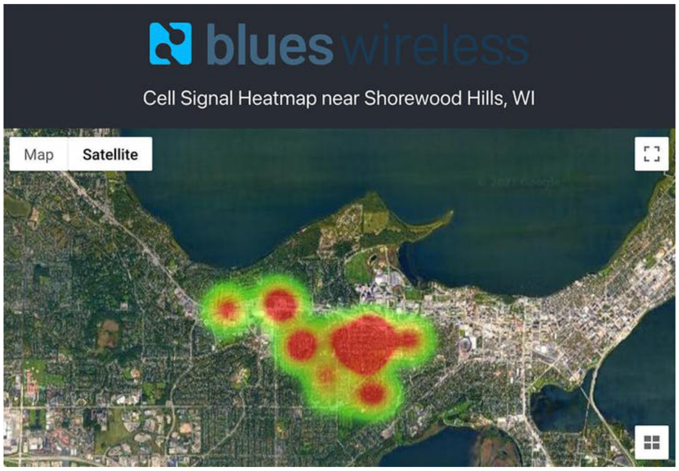
How to Build an IoT Heatmapping Device

“A heatmap overlaid on satellite imagery is a sophisticated and actionable means of displaying GPS-based data.”

Rob Lauer
Developer Relations Lead

Instructions on Hackster

Heatmaps allow you to visualize numerical data at a glance by plotting data sets on colored maps. Individual points within a data set are represented by different colors, and color variation is used to easily distinguish the magnitude of numeric values. By plotting relationships between variables, heatmaps reveal patterns in data and offer starting points for deeper analysis. These patterns can give valuable insight into business operations, such as supply chain analytics, resource utilization, wireless network signal coverage, environmental science research, eye tracking for digital assets, and more.



Cell Signal Heatmap near Shorewood Hills, WI

When building proof-of-concept or prototype IoT devices it is important to spend most of your time on features that solve business problems. To that end, Blues Wireless Notecard is the simplest, and most cost-effective way to add connectivity to IoT devices. Simply connect the Notecard to your device’s existing UART or I2C bus, and it will connect your device to the cellular network automatically, ready to transmit and receive data from its associated Notehub account.

You can build a cellular- and GPS-enabled IoT prototype to display location-based data on a heatmap by following this project. You will be able to collect wireless signal strength and location data at defined intervals. Learn how to build this cellular IoT GPS heatmapping device for less than $150, using only 32 lines of code and 6 hardware components.

Heatmapping for IoT Data Visualization

Visually displaying complex data points on heatmaps makes data analysis faster and easier. Heatmapping can be a powerful part of a company’s analytics, helping to uncover operational insights and guide strategy. By examining data patterns, businesses can measure performance in a specific area, unlock data trends, understand their market, and enable data-driven decision-making.

A heatmap overlaid on satellite imagery is a sophisticated and actionable means of displaying GPS-based data. Data is collected, stored in the cloud, and displayed on a web-based heatmap. For this type of project, you’ll use the following components:

- **MCU:** Raspberry Pi Pico
- **Wireless connectivity and GPS:** Blues Wireless Notecard and Notecarrier
- **Secure data routing:** Blues Wireless Notehub.io
- **Cloud functions, storage, and maps:** Google Cloud Platform, Google Maps, Firestore
- **Displaying in-action status updates:** 1602 LCD module (optional)

Behind the GPS and Cell Signal Heatmapping Device

Follow this project to build a device that collects cellular and location data from the Raspberry Pi Pico and Blues Wireless Notecard and maps the data using Python and Google Cloud Platform. With built-in sensors and embedded SIM, the Notecard is the quickest and easiest way to add GPS and cellular connectivity to any device. You can find the complete project assembly instructions on Hackster and the full source code on GitHub.

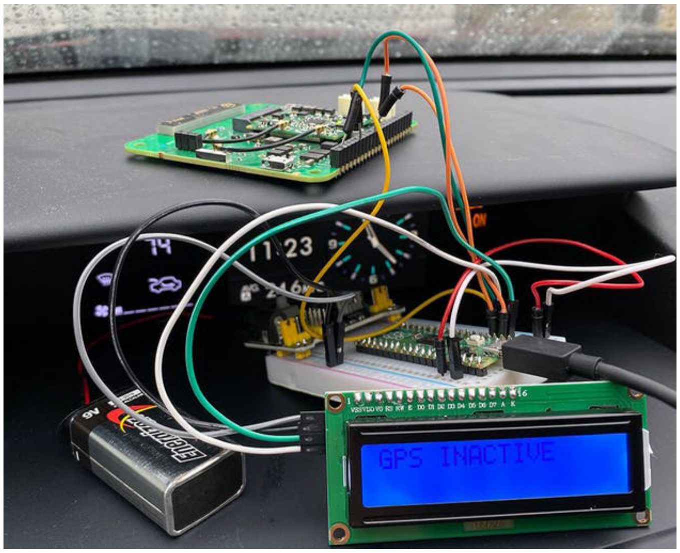
Hackster: https://www.hackster.io/rob-lauer/cellular-signal-heatmap-with-raspberry-pi-pico-and-gps-1e16df

GitHub: https://github.com/rdlauer/notecard-heatmap

Cost: $137.90 Lines Of Code: 32 Project Time: 5 Hours	Languages: • Python • JavaScript • CSS • HTML

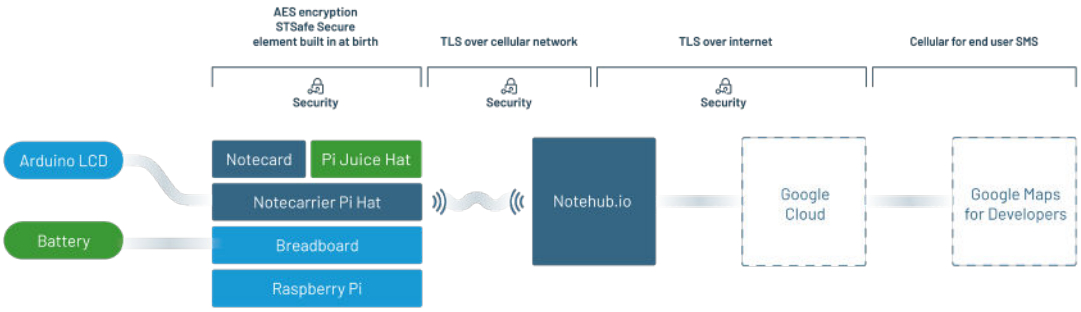
Hardware

- Raspberry Pi Pico
- Blues Wireless Notecard (Cellular)
- Blues Wireless Notecarrier-AL
- DFRobot I2C 16x2 Arduino LCD Display Module
- Solderless Breadboard Power Supply Module
- Breadboard (generic)



Software apps and online services

- Blues Wireless Notehub.io
- Google Cloud Platform
- Google Maps for Developers



The main parts of the project are:

- Assemble the hardware for your device.
- Configure the device to collect GPS data.
- Send results to the cloud.
- Create a web app for displaying the heatmap.

Displaying Your Heatmap Using Google Maps

Once your hardware is assembled and you're successfully set up to collect data and pump it to the cloud, you'll build a React web app to generate a heatmap of your data using Google Maps. It's most common to use Create React App for this, which is a simple way to create and run a React project quickly and removes the need for manual configuration. React apps are generally made up of multiple function and class components, and this one has three components:

- App.js is our root-level component.
- MapContainer.js loads the Google Maps script and provides a host container for Map.js.
- Map.js shows our map.

There are several critical aspects of this part of the project to call out:

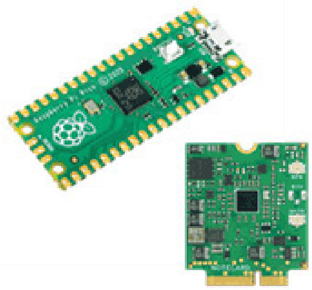
Firebase

There is some confusion between Firebase and Google Cloud Platform, and 'yarn add firebase' is an example. Both Firebase and GCP offer access to Firestore. The APIs for accessing Firestore are in the Firebase SDK. Here you'll use the Firebase SDK to access your data from Firestore.

Map Container

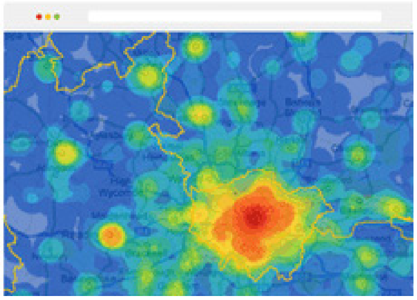
The MapContainer.js component is the host of your map and is what loads the Google Maps script. You would assume you can't use the map until the script is loaded, but the react-google-maps/api library that you'll add exposes LoadScript, which plays a key part in lazy-loading Map.js.

Gathering GPS Data



Sending Data to the Cloud

Visualizing Data on a Map



The Map

Your Map.js component exposes the core Google Map, allowing you to set the center of the map, and the HeatmapLayer exposes the Google Maps heatmap API. The data property of the HeatmapLayer is an array of all the returned data points from Firestore. A single object in the array is made up of a location (GPS coordinates) and a weight integer (cellular signal strength). The higher the weight, the more intense the heatmap color.

You can adjust the display of the heatmap by supplying options to the HeatmapLayer element. For this project you'd specify the radius, or how far you want the heatmap colors to spread around a point.

Shortcut: you can use the first record from the mapdata collection for the center.

Running the Web App

With your web app built, back in the terminal you can install dependencies with 'yarn install', and run the app with 'yarn start'. You can test your map using the 'testPoints' array in Map.js instead of the data pulled from GCP.

Now you can start collecting data and testing your map.

Applications of This Project

Heatmapping is a powerful tool for anyone who wants to easily visualize complex data sets. This type of data visualization makes data more actionable for a number of industries. Some heat map use cases include:

- Wireless network signal coverage
- Asset tracking
- Supply chain analytics
- Resource utilization (elevator usage)
- Wildlife tracking and migration patterns
- Smart city counters (pedestrian, vehicle, etc.)
- Retail shopper traffic counting
- Manufacturing production bays monitoring
- Construction site monitoring