blues wireless

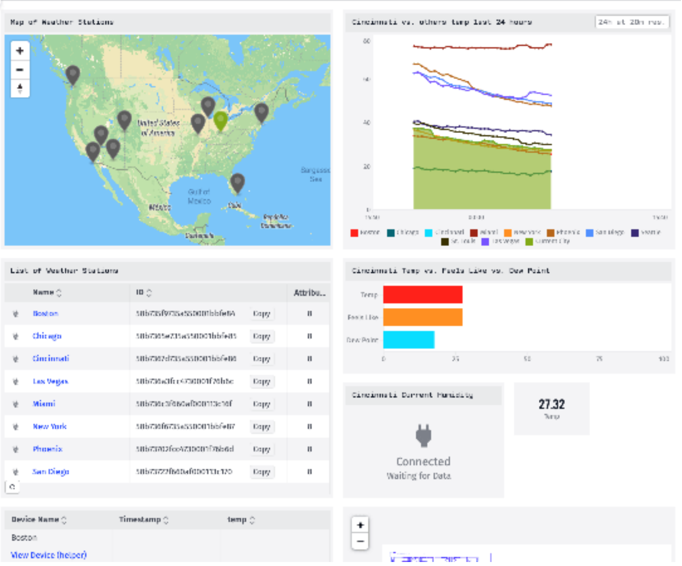
Remote Condition Monitoring with IoT and SMS Alerts

> *"By monitoring assets with IoT devices, real-time data are tracked continuously and can indicate potential issues even before they occur."*

Rob Lauer
Developer Relations Lead

Instructions on GitHub: https://github.com/bsatrom/wfh_stress_monitor

How much money do you think your business could save by predicting maintenance schedules and mechanical failures? Within a 3-year period, over 80% of manufacturing companies experience downtime with an average cost of $2 million - and that's with a downtime of about 4 hours. By monitoring assets with IoT devices, real-time data are tracked continuously and can indicate potential issues even before they occur.



Map of Weather Stations

Cincinnati vs. others temp last 24 hours

Cincinnati Temp vs. Feels Like vs. Dew Point

Cincinnati Current Humidity

Connected
Waiting for Data

27.32
Temp

List of Weather Stations

	Name	ID		Attribu...
	Boston	[illegible]	Copy	0
	Chicago	[illegible]	Copy	0
	Cincinnati	[illegible]	Copy	0
	Las Vegas	[illegible]	Copy	0
	Miami	[illegible]	Copy	0
	New York	[illegible]	Copy	0
	Phoenix	[illegible]	Copy	0
	San Diego	[illegible]	Copy	0

Device Name	Timestamp	temp
Boston		
View Device (helper)		

When building IoT device prototypes, it's best to use a Blues Wireless Notecard System on a Module. It's the quickest and most cost-effective way to add connectivity, so you can spend your time building features that solve business problems. Simply plug Notecard into an M.2 connector and it'll connect your device to the cellular network automatically, ready to transmit and receive data even in areas where Wi-Fi is unavailable.

You can build a fully functional IoT remote conditioning monitor prototype for less than $300, using only 5 hardware components.

Using IoT Devices for Predictive Maintenance

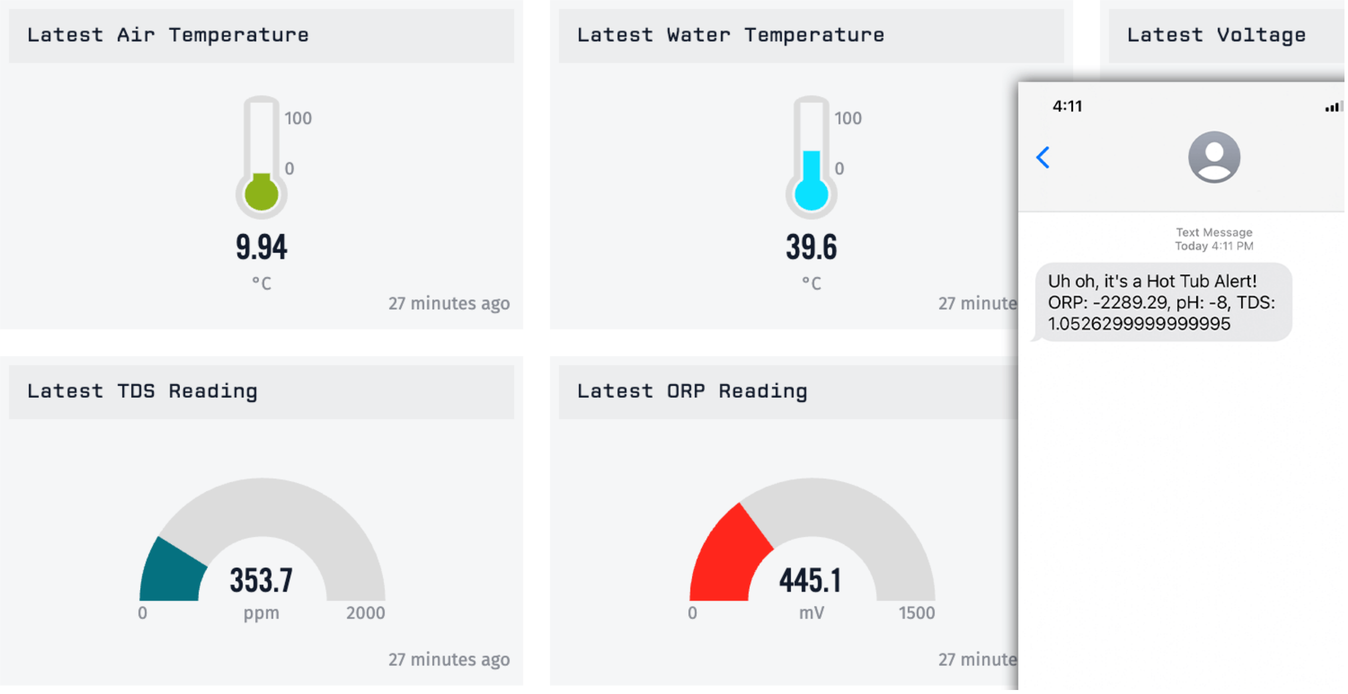
This Hackster project was built as an IoT water quality monitoring system, and the analog sensors measure pH, ORP (Oxygen Reduction Potential), and TDS (Total Dissolved Solids). However, the device can be built with different sensors to evaluate the condition of machines, equipment, and elements such as air or water. All you need to do is identify the signals of pending maintenance and detect those conditions. Some common maintenance signals are:

- Increased vibration or vibration of a particular frequency (mechanical devices)
- Presence or absence of fluid (mechanical or environmental devices)
- Air flow rate (workloads where air pressure or air flow must be maintained)
- Increased dissolved solids in suspension (fluid quality, like hot tubs)

Predictive maintenance is becoming the preferred approach over routine or time-based methods because tasks are performed only when necessary, saving unnecessary service calls or truck rolls. Plus, making repairs before something breaks is more efficient and saves money. Predictive maintenance with IoT devices:

- Avoids downtime
- Minimizes collateral damage and complexity of repairs
- Extends the life of your assets
- Maintains compliance and regulatory standards
- Improves materials and inventory management

Using the information below, this is the easiest way to build an IoT remote condition monitoring device prototype with SMS alerts.

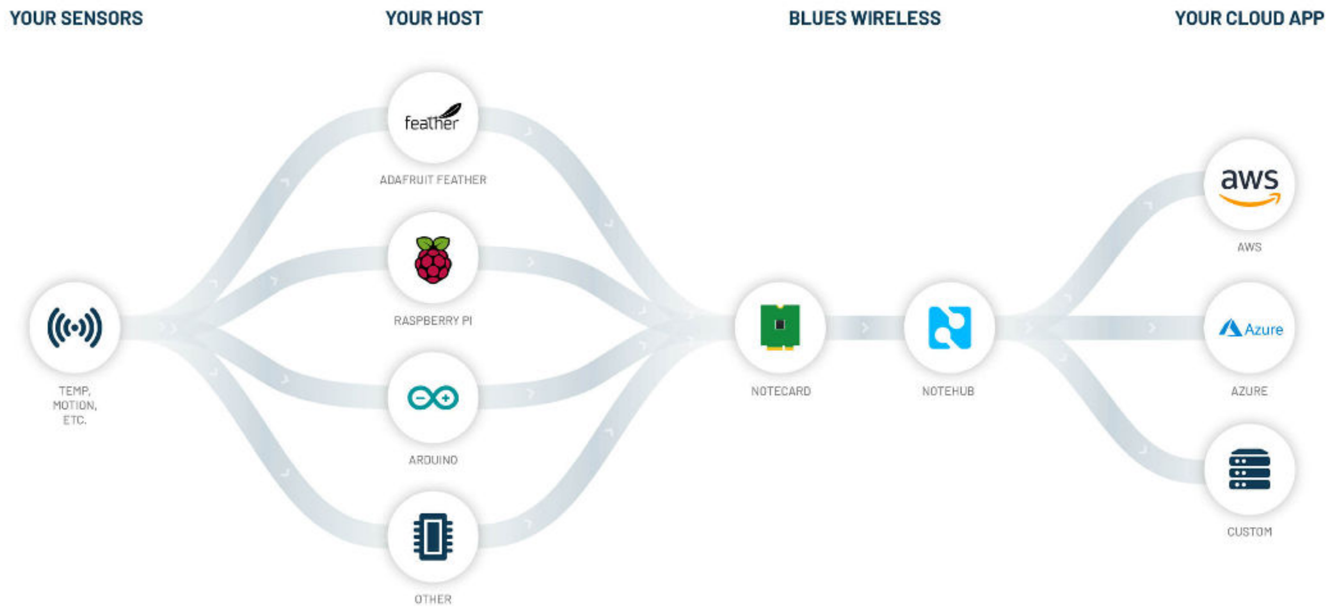


Uh oh, it's a Hot Tub Alert! ORP: -2289.29, pH: -8, TDS: 1.0526299999999995

How Blues Wireless Improves Predictive Maintenance Devices

Blues Wireless is the simplest way to add wireless connectivity to a device. In 30 minutes, you can go from unboxing to sending arbitrary data over a global cellular network, with no configuration needed. Given the many complexities in addressing predictive maintenance using IoT Devices, it makes sense to use pre-built System on a Module (SOM) technology with zero-configuration provisioning for connectivity.

In the image below, you'll see a left-to-right depiction of how sensor data moves from an edge device to a cloud application. Blues Wireless provides the infrastructure for bidirectional communication between edge devices and cloud endpoints via a combination of hardware and software. On the hardware side, in the host device, Blues Wireless Notecard provides an internal endpoint for sensor data. Notecard securely transmits the sensor data to the customer’s preferred cloud endpoint via Blues Wireless Notehub, an intermediary cloud application. Notehub provides protocol translation, transport security, data routing, device management, and device firmware update capability.



Behind the Remote Condition Monitoring Project

If you’re looking for a device that can collect time series data and send SMS alerts, this is the best project to follow. You can find the complete source code for the project at the GitHub repository linked below and the full project instructions on Hackster.

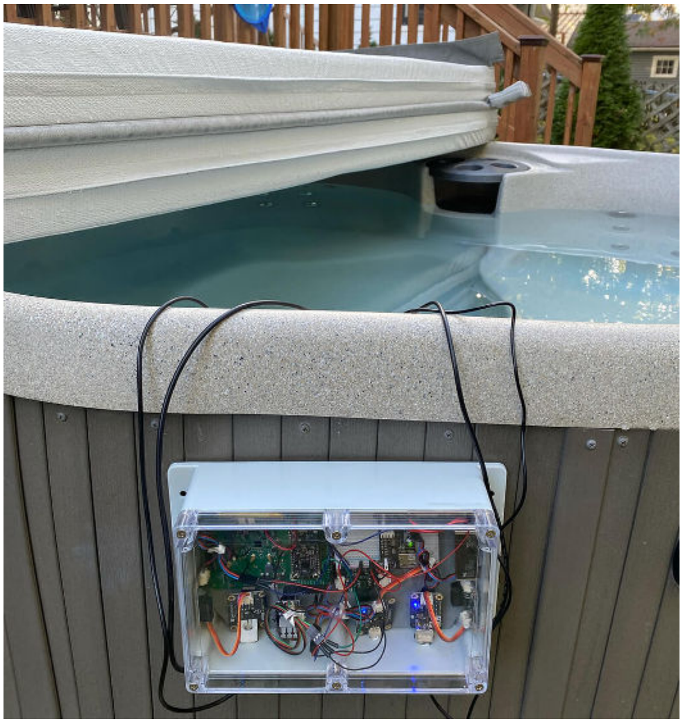
GitHub: https://github.com/rdlauer/hot-tub-time-series-machine

Hackster: https://www.hackster.io/rob-lauer/debugging-a-hot-tub-time-series-machine-92e44f

Price	$292.90	Languages	C++
Lines Of Code	30		
Project Time	6 Hours		

Hardware

- Blues Wireless Feather Starter Kit
- Blues Wireless Notecarrier-AF
- Blues Wireless Notecard
- Blues Wireless Swan
- DFRobot Gravity: Analog Spear Tip pH Sensor / Meter Kit
- DFRobot Gravity: Analog ORP Sensor Meter
- DFRobot Analog TDS Sensor
- DFRobot Gravity: Waterproof DS18B20 Sensor Kit

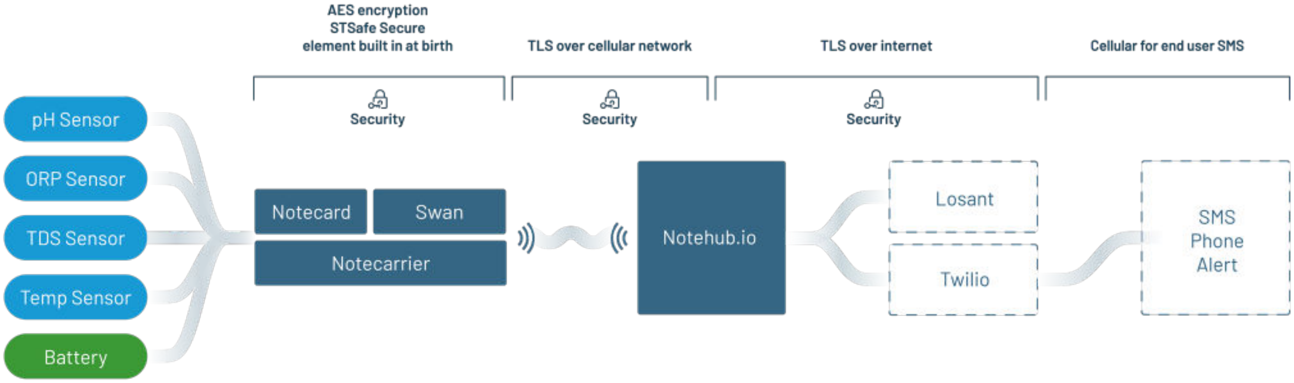


Software apps and online services

- Blues Wireless Notehub.io
- Arduino IDE
- Losant Data Visualization
- Twilio SMS Messaging API

The main parts of the project are:

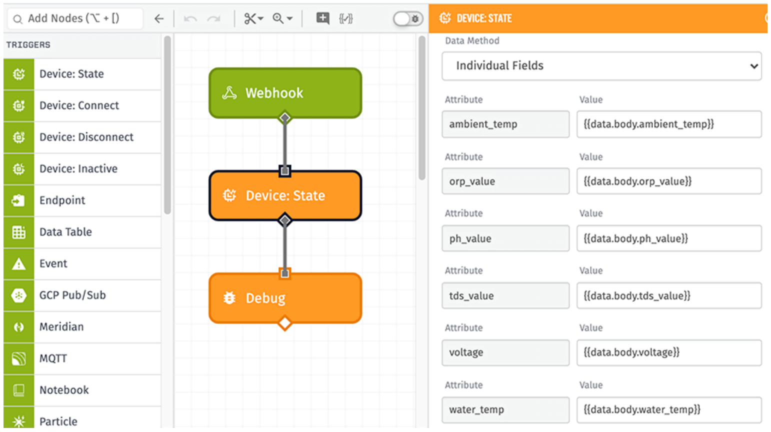
- Setting up your sensors for data collection
- Establishing alert thresholds
- Securely routing data to the cloud with Notehub.io
- Routing data as SMS alerts with Notehub.io and Twilio
- Building your dashboard



Building a Cloud-Based Dashboard

One of the more complicated parts of the process is building the cloud-based dashboard. Losant provides application templates to build your own development process. There are 5 distinct steps involved when building out a way to ingest and report data within Losant:

- Set up an Application
- Set up a Device
- Create a Webhook
- Create a Workflow
- Create a Dashboard



Data Method: Individual Fields

Attribute	Value
ambient_temp	{{data.body.ambient_temp}}
orp_value	{{data.body.orp_value}}
ph_value	{{data.body.ph_value}}
tds_value	{{data.body.tds_value}}
voltage	{{data.body.voltage}}
water_temp	{{data.body.water_temp}}

For this project, start with a "Blank Application", then add a "Standalone" device. In this case, the device is the Blues Wireless Notecard. Using JSONata, you can specify all the data attributes that will flow from the Notecard to the cloud via Notehub.io.

Then, create a webhook that triggers a workflow to process the incoming data. When the webhook is created, you are provided a "trigger URL" that serves as the base URL used in the Notehub.io route mentioned above.

Next, you'll need the workflow to process the incoming data. Using Losant's Node-RED inspired workflow engine, creating a workflow is simple:

Finally, you'll create a dashboard comprised of one or more blocks. Every dashboard is unique, and you can create blocks with gauges or displays for whatever data your sensors are collecting.

For technical support with the Blues Wireless Notecard and Notecarrier, please visit dev.blues.io.

Other Remote Condition Monitor Applications

- Detecting mosquito breeding conditions in water and triggering an agitation cycle
- Greenhouse environment monitoring
- Quality monitoring and smart metering for water utility companies
- Predicting maintenance cycles of physical equipment and machinery from microscopes to wind turbines in medical and industrial settings
- Monitoring fire hydrant pressure, municipal water quality, or garbage can fill levels in smart cities