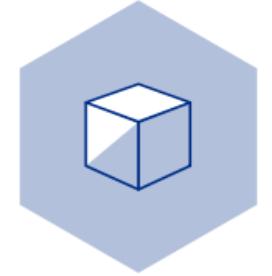




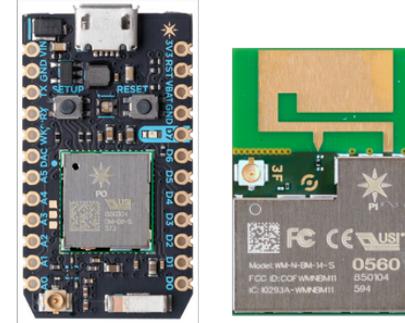
Particle

OPENTHREAD 技术研讨会

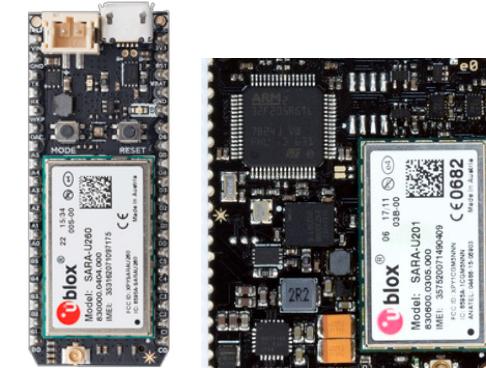
PARTICLE IS A FULL-STACK IOT DEVICE PLATFORM



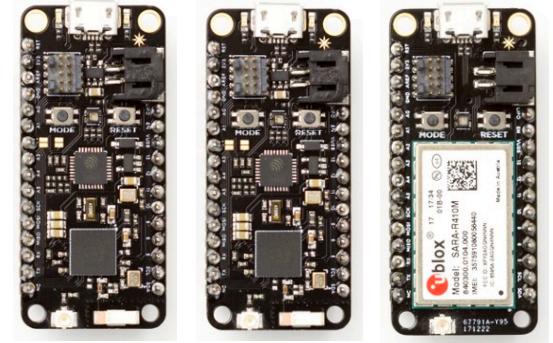
IoT Hardware



Wi-Fi



Cellular



Mesh NEW



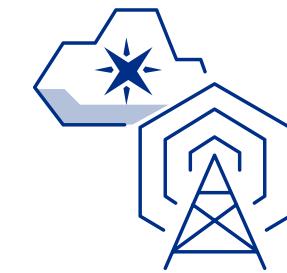
Device OS



Connectivity



2G/3G/LTE NEW
SIM Cards



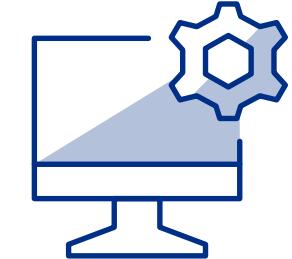
Cellular MVNO



Mesh networking NEW



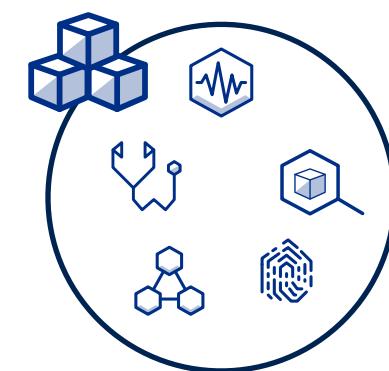
IoT Device Cloud



Developer Tools



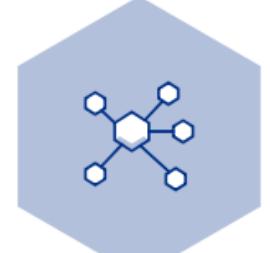
Fully managed
connectivity



Device management



OTA Firmware
Updates



IoT Apps



Integrations



SDKs

WE HAVE GROWN THE WORLD'S LARGEST IOT DEVELOPER COMMUNITY

2018

PARTICLE BY THE NUMBERS

160,000



140,000 **developers**

Developers love Particle. We have the largest IoT developer community in the industry



170 countries

Our customers' devices are deployed all over the world, from Argentina to Antarctica



8,500 companies building with Particle

According to IDC, we have the highest customer satisfaction rating of any IoT platform



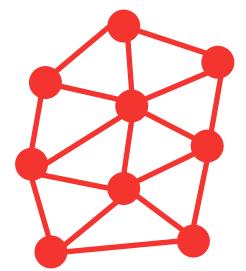
500,000 devices

We manage hundreds of thousands of devices sending billions of messages per month

CHALLENGES



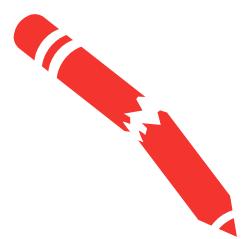
HIGH COST OF CONNECTIVITY via Wi-Fi and cellular



LARGE NUMBER OF NODES overwhelmed traditional networking solutions

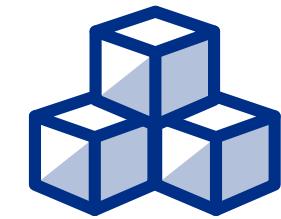


CHALLENGING ENVIRONMENTS that require flexible network and power solutions

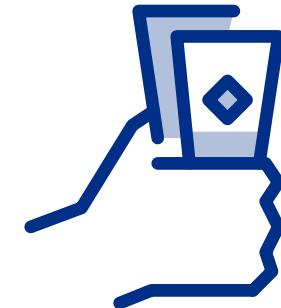


INADEQUATE TOOLS for building prototypes and solutions at scale

THE PERFECT SOLUTION WOULD ENABLE



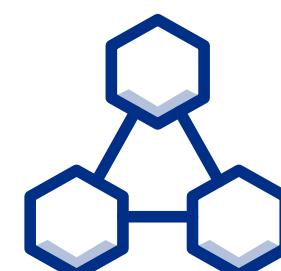
ROBUST local (device to device) communications



LOW COST incremental hardware connectivity



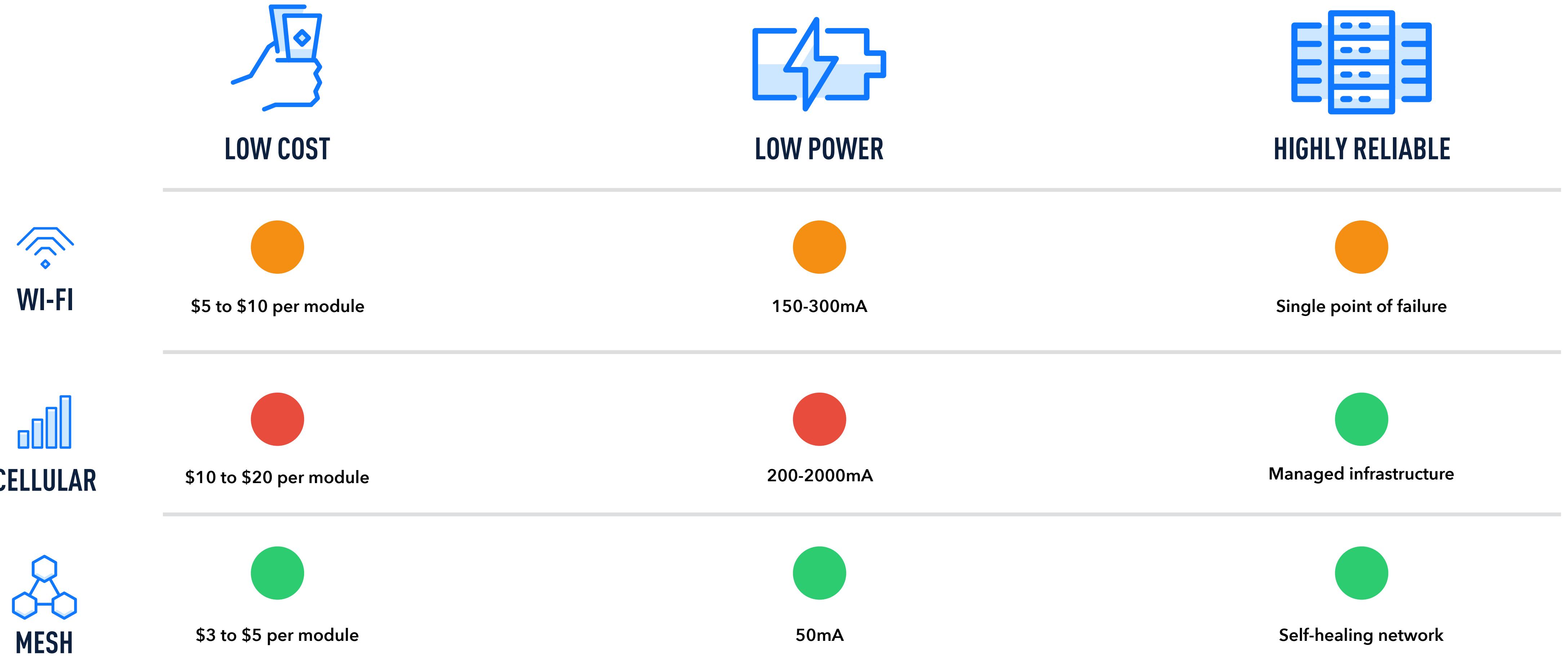
WIRELESS or battery-powered nodes



FLEXIBLE systems architecture

- » Wired/wireless
- » Cellular / Wi-Fi / Ethernet backhauls
- » Support for small and large deployments

WE STARTED TO EXPLORE MESH NETWORKING



AND THREAD WAS THE CLEAR WINNER

THREAD GROUP | Requirements

- ✓ No single point of failure
- ✓ Self-healing
- ✓ Interference robustness
- ✓ Self-extending
- ✓ Reliable enough for critical infrastructure

Requirements:

- New wireless home network
- ✓ Low power
- ✓ Resilient (mesh)
- ✓ IP-based
- ✓ Open protocol
- ✓ Secure and user friendly
- ✓ Fast time to market
- ✓ Existing radio silicon



PARTICLE IS A NEW MEMBER OF THE THREAD GROUP

A screenshot of a web browser window titled "Thread Group". The page displays a grid of company logos and names. The "Particle" logo is highlighted with a red rectangular border. Other companies listed include kirale, Kwikset, Landis+Gyr, LEEDARSON, LG, Lutron Electronics, Co., MMB Networks, nest, NXP, OSRAM, On-Bright, P&G, Qualcomm, RENESAS, runtime.io, RIGADDO, Rutledge Omni Services Pte Ltd, SAMSUNG, SALTO, SERCOM, signify, SILICON LABS, SIEMENS, SITERWELL, SOMFY, and SMK Corporation.

A screenshot of a web browser window titled "Case Studies". The page features the Thread logo at the top. The main content is a case study for "Particle".

CASE STUDIES

Particle

PARTICLE MESH: LEVERAGING THREAD TO UNLOCK REAL IOT

Particle, the most widely used Internet of Things (IoT) platform, has announced **Particle Mesh**, its next generation hardware unlocking developer access to mesh networking technology.

With Particle Mesh, networks can cover more ground and capture more data with higher reliability at a reduced cost - ultimately making IoT solutions more intelligent.

This helps developers revolutionize industries from manufacturing and logistics to smart home devices, delivering on the real IoT.

THE CHALLENGE

Historically, IoT devices connect directly to the cloud through Wi-Fi, cellular, or Ethernet communication. They depend on the cloud to relay messages between devices, even if those devices are only meters away. This approach requires all devices to be Wi-Fi, LTE, or Ethernet connected, which adds BOM and data costs, increases power demands, and introduces unnecessary points of failure leading to more network downtime.

THE PROCESS

When Particle set out to build Particle Mesh, the company was looking for a technology that defined low-level network implementation while still providing room for extensibility. Particle chose Thread because it provides a robust technical foundation to extend all the way to Particle Mesh.

Particle believes that real IoT is achievable, by providing developers with the most powerful technologies they need to solve problems. That's why Thread was selected to support Particle Mesh in its new IoT Wi-Fi and LTE developer kits. This gives developers the tools they need to succeed.

Additionally, the true impact of using open standards is enabling a community to contribute to the advancement of infrastructure that drives everyone forward. Thread was created to support developers in building connected products with a networking protocol that activates the internet in IoT.

Particle

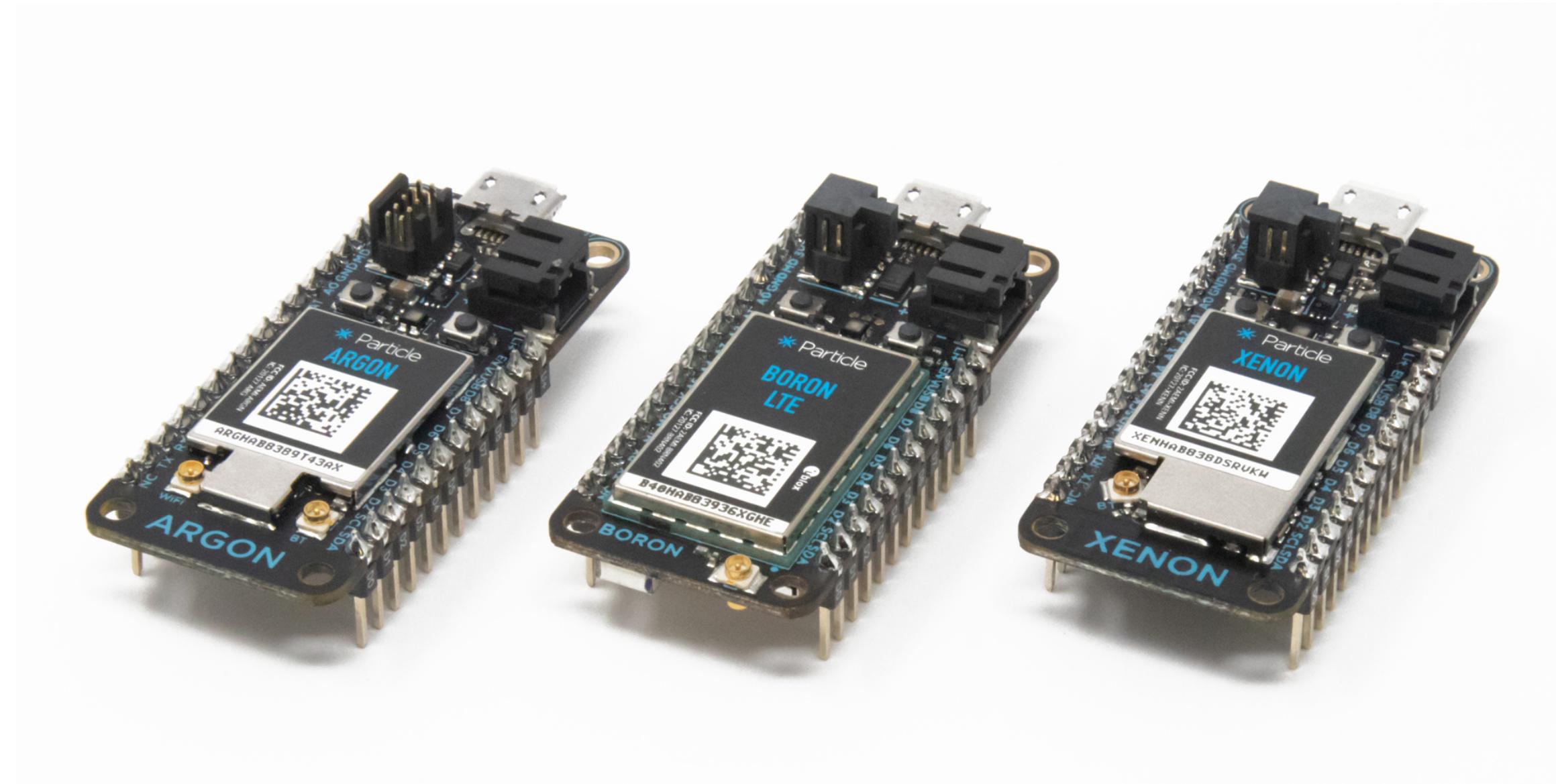
WHAT IS THREAD BUILT FOR IOT THREAD GROUP NEWS AND EVENTS SUPPORT BLOG

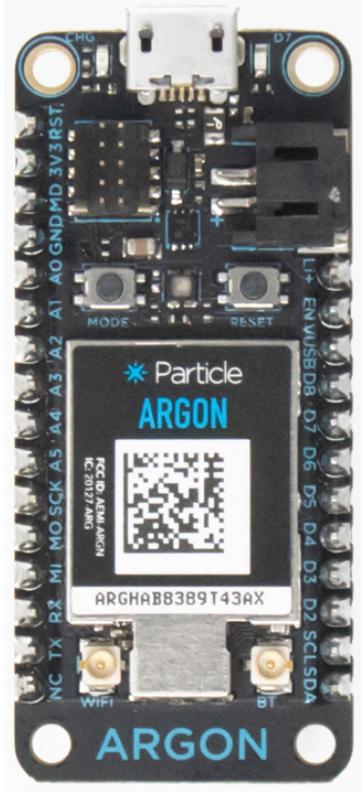
**GOOD DEVELOPMENT TOOLS WILL HELP THREAD FIND ITS KILLER
APPS BEYOND THE HOME**

in February we introduced

PARTICLE MESH

Make your IoT devices even more intelligent





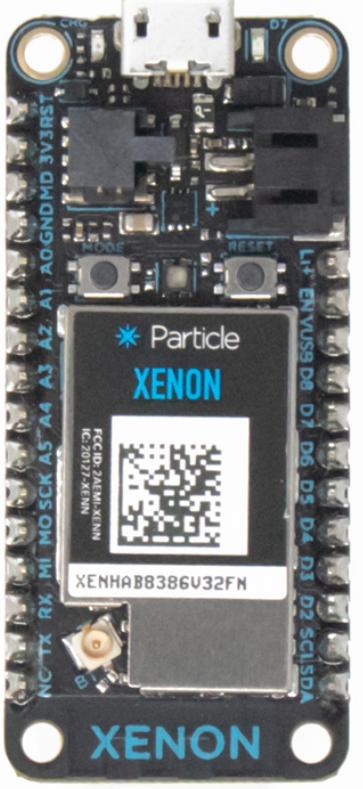
Argon

- » Wi-Fi + BLE +Mesh
- » Wi-Fi endpoint or mesh gateway
- » Starts at \$25



Boron

- » LTE-M1 or 2G/3G + BLE + Mesh
- » Cellular endpoint or mesh gateway
- » Starts at \$49



Xenon

- » BLE + Mesh
- » Mesh endpoint
- » Starts at \$15

Mesh enabled, next generation

- » Adafruit Feather form factor
- » OpenThread-based Mesh
- » Low cost
- » Low power
- » Highly reliable

INITIAL FEEDBACK WAS EXTREMELY POSITIVE

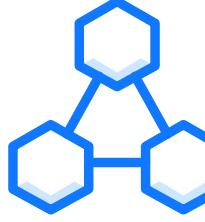
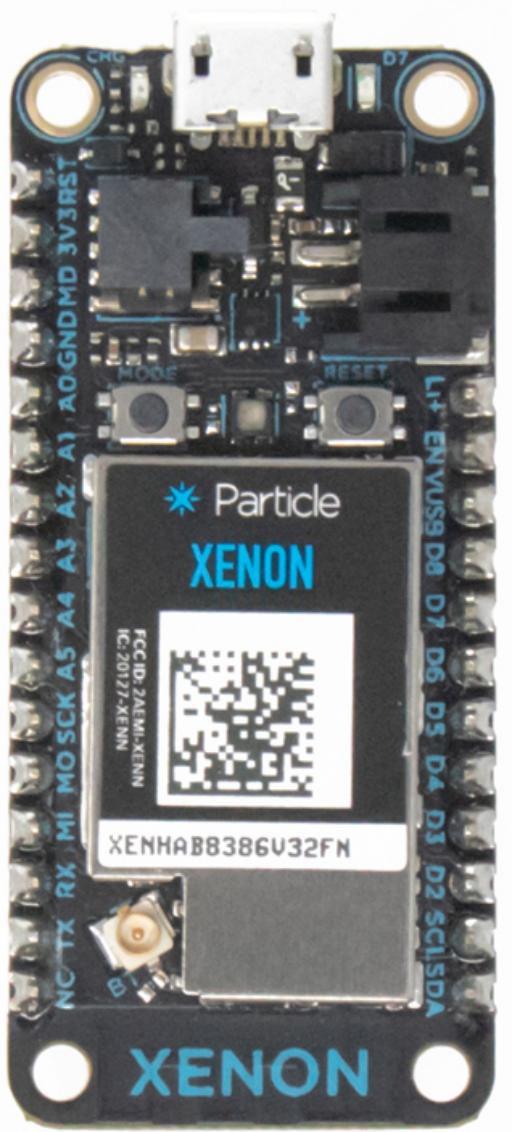
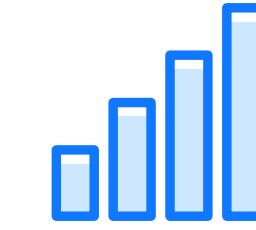
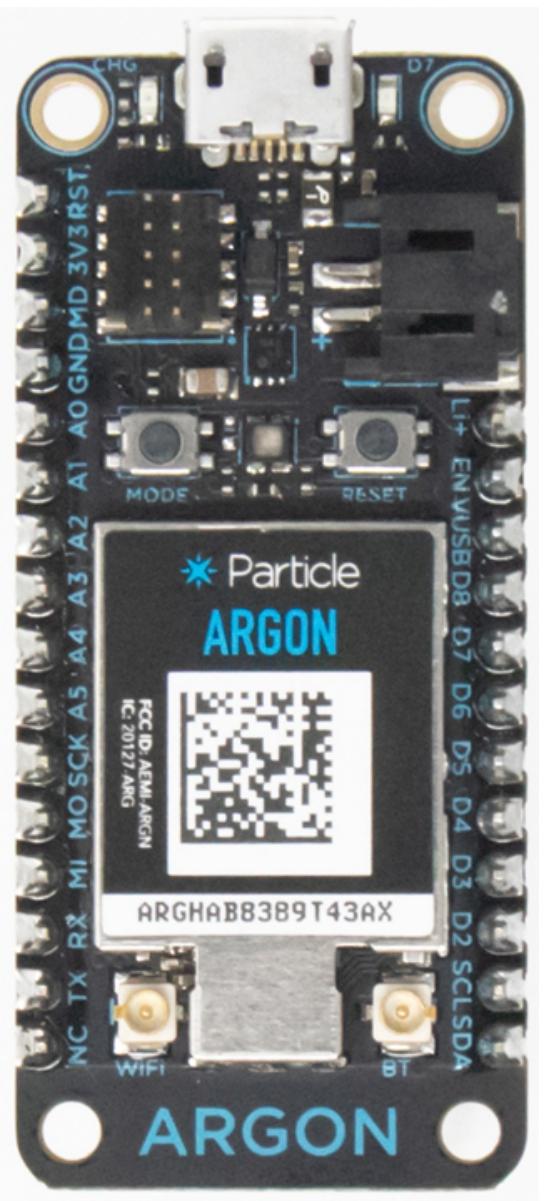
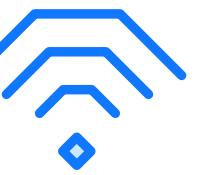
5,800 customers preordered **35,000** devices

shipment started in October



MESH BENEFITS FROM ALL THE EXISTING FEATURES OF THE
PARTICLE ECOSYSTEM

1. MASS PRODUCTION HARDWARE FOR APPLICATIONS AT SCALE



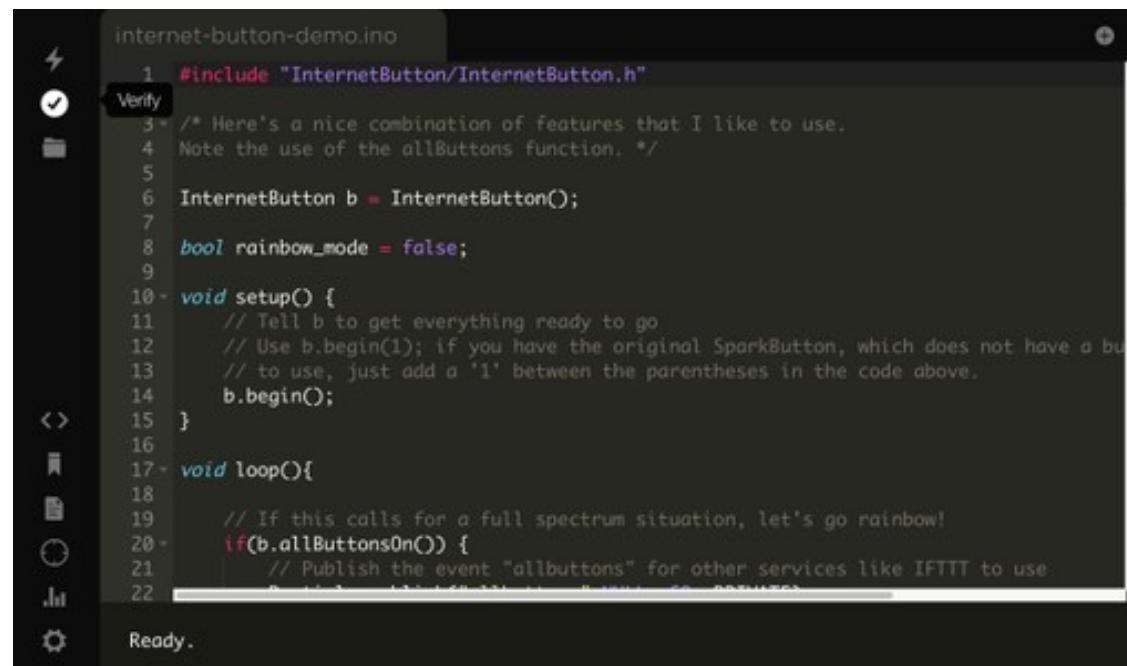
Wi-Fi

Cellular (2G/3G/LTE)

Mesh

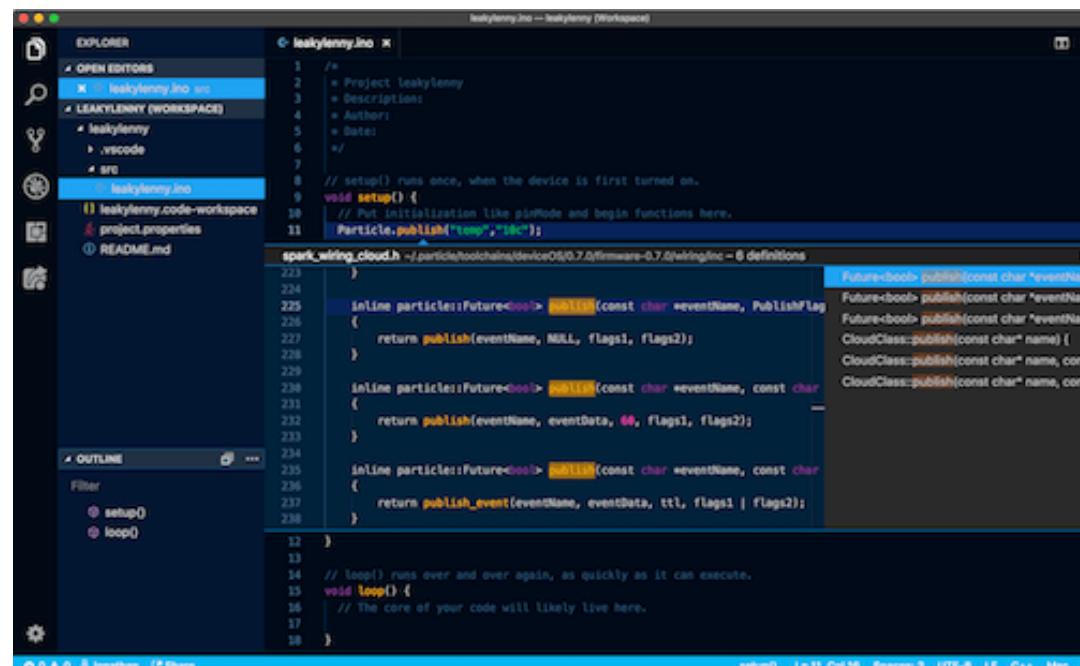
2. PRE-CONFIGURED TOOLS THAT ACCELERATE PRODUCT DEVELOPMENT

INTEGRATED DEVELOPMENT TOOLS



```
internet-button-demo.ino
1 #include "InternetButton/InternetButton.h"
2
3 /* Here's a nice combination of features that I like to use.
4 Note the use of the allButtons function.
5
6 InternetButton b = InternetButton();
7
8 bool rainbow_mode = false;
9
10 void setup() {
11     // Tell b to get everything ready to go
12     // Use b.begin(1); if you have the original SparkButton, which does not have a button
13     // to use, just add a '1' between the parentheses in the code above.
14     b.begin();
15 }
16
17 void loop(){
18     // If this calls for a full spectrum situation, let's go rainbow!
19     if(b.allButtonsOn()){
20         // Publish the event "allbuttons" for other services like IFTTT to use
21         Particle.publish("allbuttons", "true");
22     }
}
```

WEB IDE

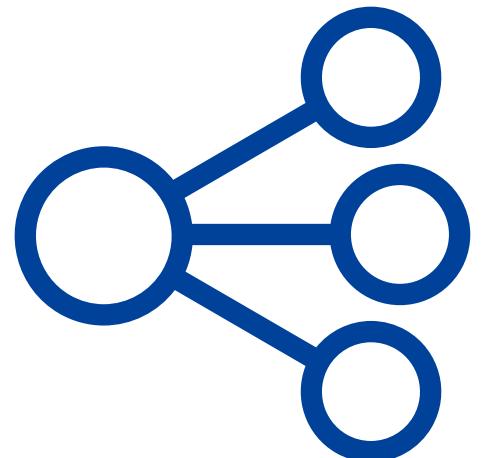


```
InternetButton.ino
1 // Project: testkeyenny
2 // Descriptions:
3 // Authors:
4 // Date:
5
6
7 // setup() runs once, when the device is first turned on.
8 void setup() {
9     // Put initialization like pinMode and begin functions here.
10    Particle.publish("hello", "world");
11
12    // loop() runs over and over again, as quickly as it can execute.
13    void loop() {
14        // If this calls for a full spectrum situation, let's go rainbow!
15        if(b.allButtonsOn()){
16            // Publish the event "allbuttons" for other services like IFTTT to use
17            Particle.publish("allbuttons", "true");
18        }
19    }
}
```

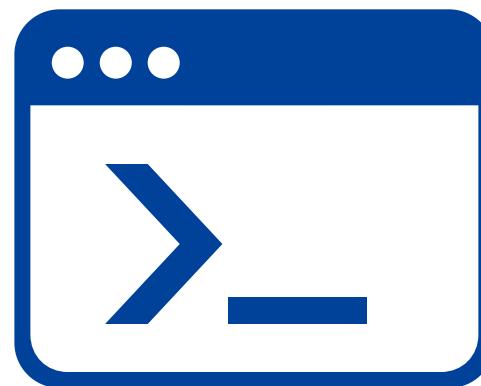
DESKTOP ENVIRONMENT

API / SDKS / CLI

REST API



CLI FOR AUTOMATION



WEB/MOBILE SDKs



BUILT IN OTA UPDATES

3,000+ FIRMWARE LIBRARIES

C/C++ OR ARDUINO (WIRING)

SECURE BY DEFAULT

3. SOFTWARE STACK THAT IS OPEN, FLEXIBLE, AND FREE TO USE

OPEN THREAD

released by Nest



BUILT ON OPENTHREAD

FREE TO USE



Microsoft Azure



Your business already relies on dozen of services to operate effectively. Device Cloud supports integrations so you can seamlessly connect device data with the tools your business relies on, saving you time and money.

DEVICE CLOUD INTEGRATIONS



WE ALSO ADDED KEY FEATURES TO SUPPORT THREAD
DEVELOPMENT

TURNKEY CLOUD THAT WORKS WITH THREAD

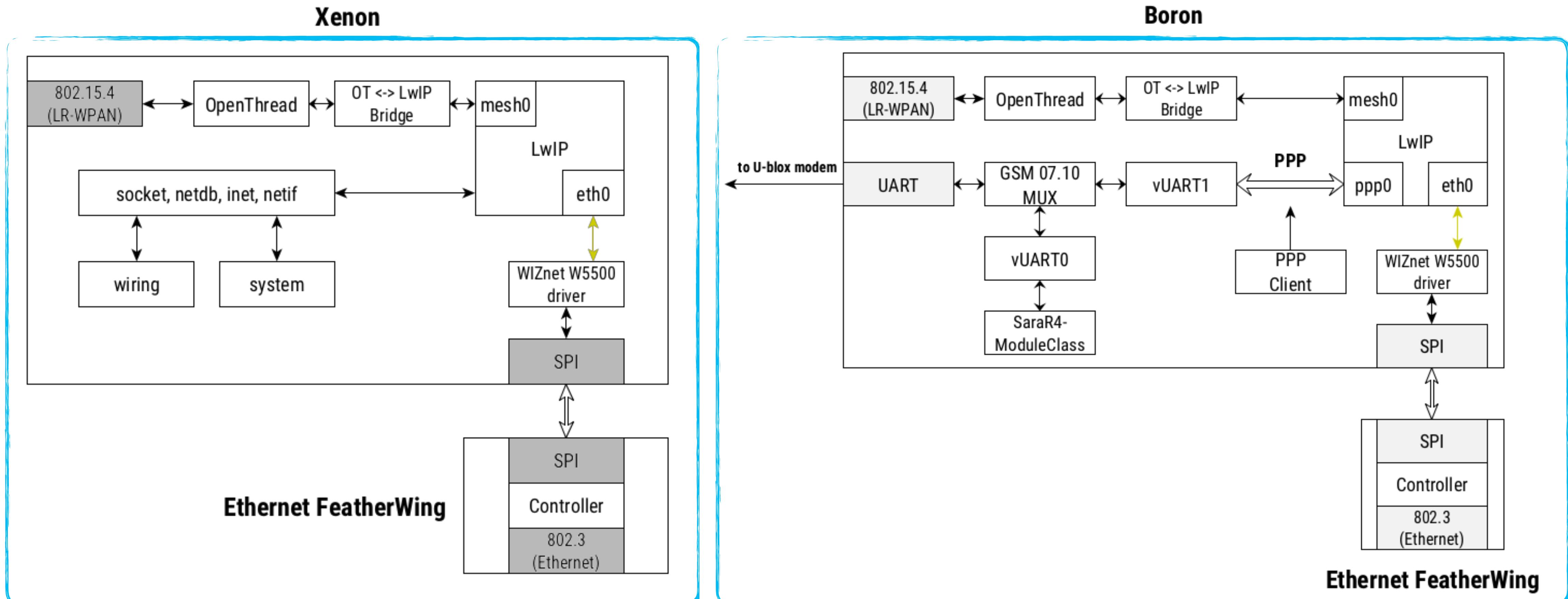


The screenshot shows the Particle Cloud interface with several overlapping windows:

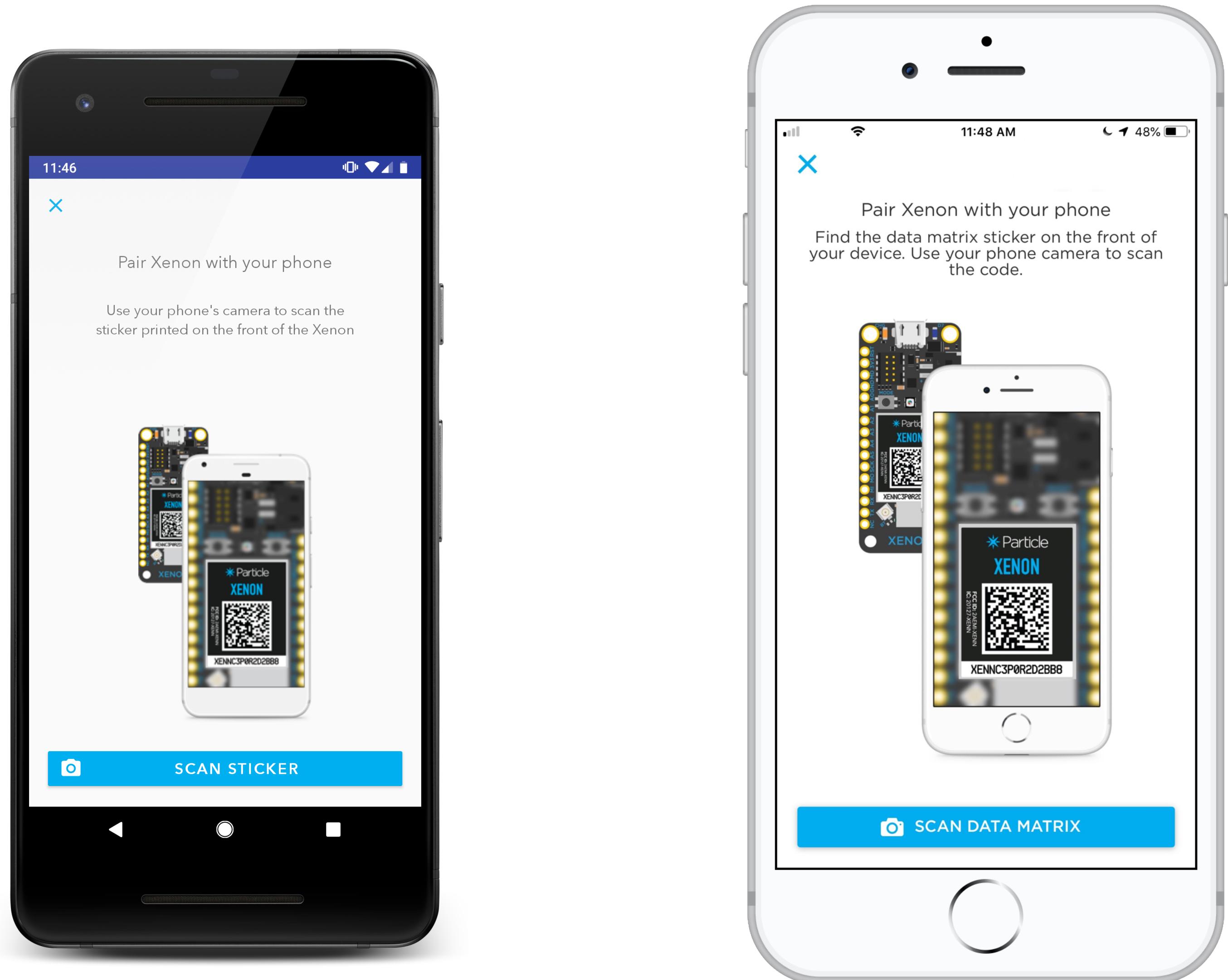
- Devices**: A list of connected devices with IDs like 42003, 3b002, 39003, 36004, 36002, 36002, 33003, 2f001e, 2e004, and 2b004.
- Firmware**: A section showing "Version 2" by "Jeff@particle.io" uploaded an hour ago. It includes an "UPLOAD" button.
- Integrations**: A placeholder section for integrating the device with other services.
- Logs**: A chart showing event logs over time. The chart has two y-axes: one for temperature (0-15) and one for humidity (0-15). Data series include temperature, humidity, digitalWrite, analogRead, digitalRead, device went offline, and device came online. Below the chart is a table of log entries:

EVENT NAME	DATA	PUBLISHED AT	DEVICE
humidity	81	August 31st at 5:37:56 pm	3d0042000a47...
temperature	95	August 31st at 5:37:56 pm	3d0042000a47...
humidity	80	August 31st at 5:37:53 pm	3d0031000a473...
temperature	93	August 31st at 5:37:53 pm	3d0031000a473...
humidity	80	August 31st at 5:37:52 pm	3d0042000a47...
temperature	93	August 31st at 5:37:52 pm	3d0042000a47...
humidity	78	August 31st at 5:37:49 pm	3d0031000a473...

MCU BORDER ROUTERS SUPPORTING WI-FI, CELLULAR (2G/3G/LTE M1/NB1) OR ETHERNET



BLUETOOTH SETUP, INSTALLATION AND DIAGNOSTICS



SIMPLIFIED LOCAL COMMUNICATION

Firmware

▼ Cloud Functions

- Particle.variable()
- Particle.function()
- Particle.publish()
- Particle.subscribe()
- Particle.unsubscribe()
- Particle.connect()
- Particle.disconnect()
- Particle.connected()
- Particle.process()
- Particle.syncTime()
- Particle.syncTimeDone()
- Particle.syncTimePending()
- Particle.timeSyncedLast()
- Get Public IP
- Get Device name
- Get Random seed
- ▶ WiFi
- ▶ SoftAP HTTP Pages

Cloud Functions

Particle.variable()

Expose a *variable* through the Cloud so that it can be called with `GET /v1/devices/{DEVICE_ID}/{VARIABLE}`. Returns a success value - `true` when the variable was registered.

Up to 20 cloud variables may be registered and each variable name is limited to a maximum of 12 characters.

Note: Only use letters, numbers, underscores and dashes in variable names. Spaces and special characters may be escaped by different tools and libraries causing unexpected results.

It is fine to call this function when the cloud is disconnected - the variable will be registered next time the cloud is connected.

Prior to 0.4.7 firmware, variables were defined with an additional 3rd parameter to specify the data type of the variable. From 0.4.7 onward, the system can infer the type from the variable's initial value and the variable address.

// EXAMPLE USAGE

```
int analogvalue = 0;
double tempC = 0;
char *message = "my name is particle";
String aString;

void setup()
{
    // variable name max length is 12 characters long
    Particle.variable("analogvalue", analogvalue);
    Particle.variable("temp", tempC);
    if (Particle.variable("mess", message)==false)
    {
        // variable not registered!
    }
    Particle.variable("mess2", aString);

    pinMode(A0, INPUT);
}
```

PARTICLE WORKBENCH

A screenshot of the Particle Workbench interface, which is a Visual Studio Code extension. The main window shows an open workspace named 'leakylenny'. The 'EXPLORER' pane on the left lists files like 'leakylenny.ino', '.vscode', and 'spark_wiring.cloud.h'. The 'CODE' pane displays a C++ file 'leakylenny.ino' with code including `Particle.publish("temp", "10c");` and a completion dropdown for the 'publish' function. The 'OUTLINE' pane at the bottom shows symbols for 'setup()' and 'loop()'. The status bar at the bottom indicates 'setup()' is active, with 'Ln 11, Col 17' and other build-related information.

Designed for IoT, optimized for Particle

- » Local compilation
- » Full integration with Visual Studio Code
- » Managed toolchain and OTA
- » Autocomplete with IntelliSense
- » 3,000+ official Particle libraries
- » 8,000+ Visual Studio Marketplace extensions
- » Advanced debugging

Sign up to access the developer preview
www.particle.io/workbench

OPEN SOURCING

Device OS, development tools
& the hardware designs of our Dev Kits are all open source

DEMO

GETTING STARTED

PARTICLE DOCS

<https://docs.particle.io>

Particle Mesh Workshop

<https://part.cl/workshop>

getting more hardware

Particle online store or SeeedStudio

CAT M1 & NB-IOT SUPPORT

China - China Mobile (NB-IoT)

Hong Kong - 3HK (NB-IoT)

Singapore - Singtel (CAT M1 & NB-IoT)

USA - AT&T (CAT M1)

China Unicom - start testing soon

China Telecom - must operator own server + services

OUR GOAL

Help make Thread the premier mesh networking technology for IoT use cases beyond the connected home

THANK YOU!