Case study

Developing Software for a Drone Battery Charging and Data Management Unit

Charging drone batteries along with sorting and copying data produced during a whole day of shooting is a pain for all drone pilots. Our client decided to make their lives easier by creating a device that manages batteries and copies data from a drone's SD card to cloud storage. To develop software for drone battery charging, our client needed a team with experience in embedded and mobile app development, virtualization, and reverse engineering.

After studying the market, they decided to trust their project to the Apriorit team. Explore this case study to find out how we helped our client create a stable device MVP.

The client

Our client is a company that delivers an automatic drone battery charging system for aerial photography, and they were looking for an experienced engineering team to develop software for it. The system had to charge up to six drone batteries simultaneously and help drone pilots efficiently manage their photos and videos.

The challenge

Drone owners needed to be able to control the drone battery charging process, manage data on a drone’s SD card, and upload that data to the correct endpoints.

The main requirements for the project were the following:

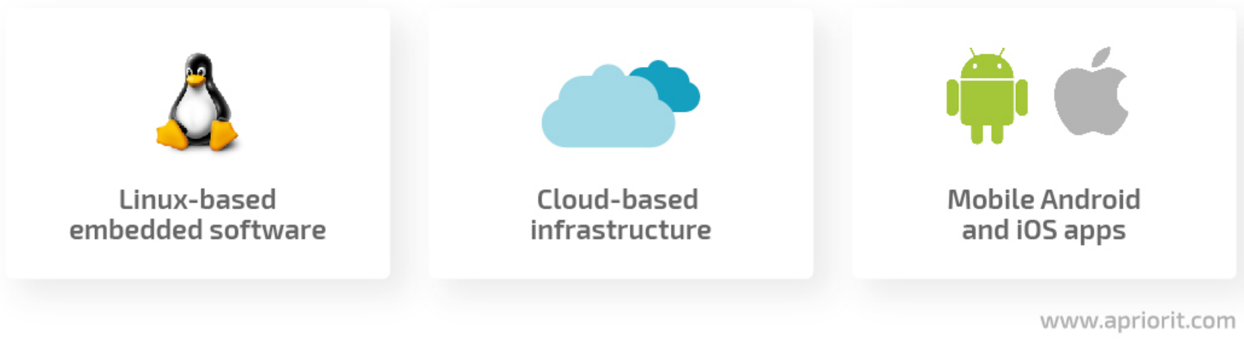
- Automatically define the type of battery and monitor its power condition
- Display and control the battery charging process from mobile apps
- Upload data from a drone’s SD card to AWS instances
- Allow for interaction with the charging kit via Wi-Fi, BLE, and LTE
- Preview recorded media from the SD card in mobile apps
- Provide support for automated over-the-air updates

Our approach

To develop the drone battery charging system, Apriorit provided the client with a dedicated team to:

1. Create Linux software and firmware for a single-board computer that serves as a hub for the client’s device
2. Develop cloud-based infrastructure to support the system
3. Deliver mobile apps for Android and iOS that allow drone pilots to control the system

KEY COMPONENTS OF THE CLIENT'S PROJECT


Linux-based embedded software
Cloud-based infrastructure
Mobile Android and iOS apps
www.apriorit.com

After assessing several outsourcing development teams, our client chose to work with us. This project required a diverse development team with experience in various technologies. For example, we needed to reverse engineer batteries to get data about device type and charge state. We also needed virtualization skills to develop software without a physical charging kit.

To bring together the required skills, we included various specialists on the project team:

APRIORIT DEDICATED TEAM			
Business analyst	Project manager	Linux developers	Mobile developers
DevOps engineers	Backend engineers	Reverse engineer	QA specialists

KEY TECHNOLOGIES	
IOS	Swift Technologies, Core Bluetooth, Amazon AWS SDK (Cognito, API Gateway), Dropbox SDK, JSON, Alamofire, KIF
Android	Java, Amazon AWS SDK (Cognito UserPools), ButterKnife
Linux	C++, Python, Bluez, GObject, grpc, D-Bus, CMake
Backend	JS, serverless config

www.apriorit.com

The result

We developed an MVP of the drone battery recharging kit and received positive feedback from the client. The MVP includes embedded software for the single-board computer, an iOS application, and cloud infrastructure to support the system.

Our client has received positive feedback on the MVP from drone pilots during testing.

How we did it

To plan the work on this project, we divided it into four key stages:

KEY PROJECT STAGES

STAGE 1	Project discovery	STAGE 2	Proof of concept (PoC)
STAGE 3	Demo alpha version	STAGE 4	Minimum viable product (MVP)

www.apriorit.com

1. Project discovery

Before drone battery charging system development, our business analysts and developers conducted a thorough project discovery. During project discovery, we studied how single-board computer software communicates with mobile apps and hardware elements. This knowledge helped us ensure that our solution would upload data from the charging unit to the cloud. We also decoded and identified battery packs and used this information to allow our software to determine the types of batteries connected and their charge levels.

After collecting all information about this project and discussing it with the client, we were ready to start development.

2. Proof of concept (PoC)

Since we didn't have a physical production unit of the client's device, we created a Linux virtual machine (VM) that simulated the behavior of the Pine ROCK64 platform. Using a VM allowed us to start development before receiving a real device. It also allowed the client to adjust their drone kit to our software.

We enabled software for charging drone batteries to:

- Identify and manage drone batteries
- Communicate with the cloud to upload data to an AWS Simple Storage Service instance
- Index, copy, and delete data from SD cards

At this stage, we decided to focus only on the iOS application to manage drone battery charging and deliver the Android version later. This saved us many development hours, since we were fixing and improving only one app instead of two.

The developed iOS application was able to:

- Communicate with a client's device via Bluetooth
- Connect with Dropbox and DroneDeploy
- Display battery statuses
- Manage user profiles
- Send push notifications

We heavily tested the embedded software and iOS application to make sure they worked flawlessly. Our client gave positive feedback on the PoC, so we proceeded to the next development stage.

3. Demo alpha version

When working on the demo, we focused on enhancing embedded software for the Pine ROCK64 platform and the mobile application. We added more software and battery management options for the client's device, such as:

- **Over-the-air updates.** We made the device able to download and install automated over-the-air updates for small components like daemons and applications. To achieve this, we implemented a system firmware update scheme that supports dual boot.
- **Automated battery management.** When a user plugs a battery into the drone charging system, it automatically detects the battery type, cell count, voltage, health, and status. After that, the kit starts charging the battery and sends the collected information to users' phones.
- **Device monitoring.** The software monitors a device's internal temperature and overall power draw. We also added a software integrity check to enable a device to detect suspicious software manipulations.

We improved the iOS application as well. In the demo version, the app could:

- Connect to the device via Bluetooth, Wi-Fi, and LTE
- Manage battery charging and notify users on its completion
- Allow users to choose where to upload data

4. Minimum viable product

To deliver the MVP, we only needed to port our software from the VM to the physical target device. However, this was one of the most challenging stages of the project because we discovered a lot of issues with the hardware. For instance, it would lose the Wi-Fi connection and had issues with power management. Also, there was no galvanic isolation of electrical sections on the single-board computer. Because of this, some board components came unsoldered with a 70% to 80% load.

We worked with the client to fix these issues and deliver a stable MVP. We also reported our discoveries to hardware manufacturers and searched for third-party experts that could help us out.

Additionally, we developed an Android application to manage the drone charging process and upload media from an SD card to cloud endpoints. The interfaces and features of this application were identical to those in the iOS app that we'd already developed.

The impact

Our client got a stable MVP that they can try out with drone pilots in order to gather feedback on how to improve the device. We also delivered iOS and Android applications to manage the charging process and upload data to the cloud. AWS-based cloud infrastructure supports this system. When our client is ready to develop the final version of the device, we’ll be here to help them.

**Have a similar project that needs diverse development expertise?
Contact us to discuss how we can help!**