A Developer's Guide

7 Considerations for Successful Development of Your IoT Device

Accelerate Cellular Development By Addressing These Key Elements Upfront

NimbeLink®
Smart. Simple. Cellular.

Summary

In this document, the experts at NimbeLink share their recommendations for successfully traversing the IoT cellular development landscape. Every embedded cellular design should begin by exploring these seven key considerations. Doing so will accelerate your cellular development, enable compatibility throughout the system, and ensure that your design will reliably deliver the data that your use case demands.

This is not intended to be a technical deep dive into any one area. To provide more detail, we have provided links at the end of the document that will take you to how-to information on each topic. The information included in this document will apply to all cellular development programs whether you're using end-device certified Skywire® embedded modems or another solution.

Note: If you are using end-device certified Skywire modems in your cellular development, you will entirely avoid the carrier certification process. If not, you will need to pursue carrier certification for your end-product. This can require a considerable amount of time and money and should be factored into your workflow.

Table of Contents

Consideration 1

Selecting Your Antennas

It sounds simple, but it's crucial that you understand your use case and your antenna requirements before you begin development. If you know that your device is going to operate over a wireless network radio frequency (RF), then you need to research and define the antenna(s) you will use at the earliest stages of your device design. In general, your antenna selection will hinge on two factors:

1. The space you have available for your antenna
2. The frequency range of the antenna, which must match the frequencies for the network you will use

For IoT devices involving LTE Category 1 (Cat 1) and up, you will need to design-in two antennas that are matched to the frequencies you'll be using. One of these will be the Primary Antenna, and the other the Diversity one. When using Skywire modems for connectivity, the Primary Antenna will be used for transmit and receive, while the Diversity one will improve overall reception. For LTE-Category M1 (Cat M1 or LTE-M) and NarrowBand-IoT (NB-IoT), you will need only one cellular antenna.

As a general rule of thumb, the closer the antenna will be to your board, the more you need to take the antenna into account as you design your custom printed circuit board (PCB). For example, if you will have a trace antenna, or one directly on your PCB, then the antenna is an integral part of your design and good RF engineering practices are crucial to the success of your device. If the antenna is inside your enclosure, but not physically on your board then care must be taken for its placement, but it's much easier than an antenna mounted on your board.

Continued on Page 5

And if your antenna will be mounted to the outside of your enclosure and attached to your board with a cable, that is the easiest option for good performance. No matter what antenna type is used, don't forget to include the required ground plane in your product. The ground plane specifications can be found in the data sheet of your antenna. And, in all cases, you must ensure that your antenna is correctly impedance matched.

Review data sheets from antenna manufacturers to determine which ones match your frequencies and your desired form factor. Some manufacturers have filters on their websites that you can use to zero-in on the best solution for your device.

If you plan to use a Skywire modem in your device, you can find a selection of some of the known compatible antennas in the documentation section of the NimbeLink website, or by contacting our **Product Support team**. When using a Skywire modem, choose an antenna with a U.FL connector, or use a U.FL to SMA adapter to facilitate the use of other types of antennas.

Key Questions

- On which frequencies will my antenna need to operate?
- How many antennas will I need?
- What antenna form factor is best for my device?

"In general, your antenna selection will hinge on two factors: the space you have available for your antenna, and the frequency range of the antenna, which must match the frequencies for the network you will use."

Consideration 2

Logging Your AT Commands

Almost every IoT device will fail at some point for some reason, so designers should implement fallback mechanisms that enable recovery from failures. For this reason, logging AT commands and their responses is essential. Doing so provides a record of every communication with your modem, and helps you diagnose and remedy a problem that you might encounter down the road.

An AT command, or attention command, is an instruction to your modem to take a specific action. Logs are, in essence, recordings of the conversations with your modem. By pulling up the commands that were sent to your modem and examining the responses to those commands you'll know exactly what is happening with your device should a problem arise. Among other things, logs can reveal problems with firmware, response timing, and signal strength.

All AT command logs are stored on your device, and you can set parameters for what commands to log and how long you want to keep them. The rolling log will then overwrite older entries so it uses up less storage space.

How long should you store logs before they are overwritten? The answer will depend on your use case and how crucial past logs will be in defining a problem. If you want to retain your logs for a long period of time, you might want to add another RAM or FLASH chip on your board to accommodate this storage.

Key Questions

- Which of my AT commands will I log?
- How long will I store specified AT commands in my log?

Consideration 3

Creating a Network Monitor

A Network Monitor is simply some code that enables you to periodically and automatically check your network connection to see if it is working properly. If the modem does not respond appropriately to the monitoring queries, the network monitor can then take steps to resolve the problem, such as getting the device reconnected, restoring appropriate settings, or even resetting the device to get it back on track.

At the most basic level the network monitor might issue AT commands to the modem to ensure that it is connected to the network. Or your network monitor might ask what the signal strength is at a point in time. Or it will send queries to determine if the sensors are properly reporting in, or if your modem is frozen or your code is stuck in a loop.

A network monitor is written to detect and resolve issues, and in each case, it can be written to throw a note in the log so you have a record of its activity. Your network monitor can be as detailed and as active as you want it to be. Generally speaking, the more critical the function of your device, the more robust your monitoring should be.

Key Questions

- What should my network monitor watch for?
- What actions should my network monitor take if it encounters specific problems?
- How active should my network monitor be – constantly running or checking in on a preset frequency?

“Generally speaking, the more critical the function of your device, the more robust your monitoring should be.”

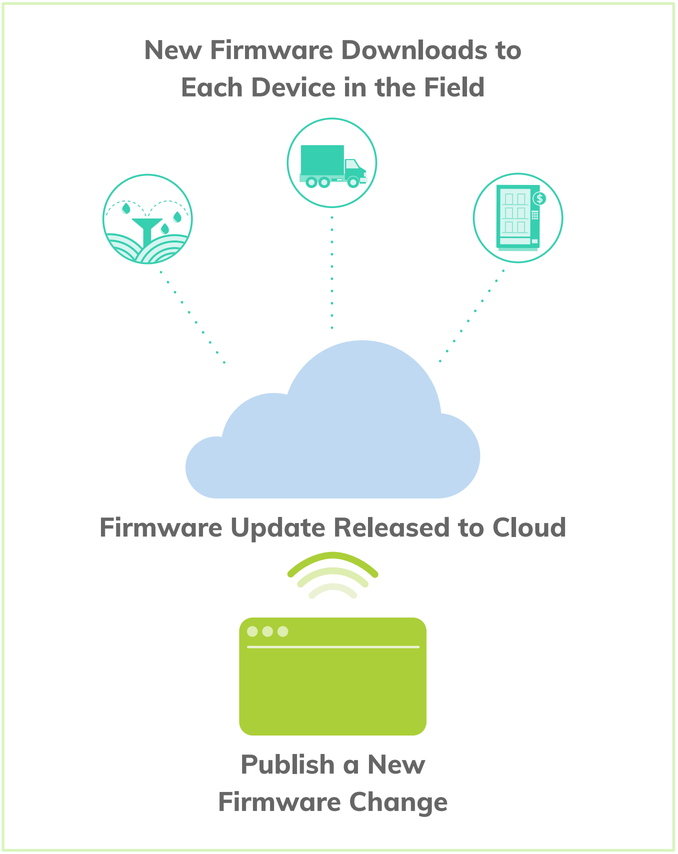
Consideration 4

Enabling Firmware Over-the-Air (FOTA) Updates for the Device

Today, software development for connected devices is not a single event, but rather an ongoing process in which new features and security patches can be added to the devices' firmware even after they have been deployed. This enables continuous improvement of the IoT product and can extend its useful life.

Similarly, if something goes wrong with your deployed IoT devices, you'll want to have the ability to remotely address the problem without sending a truck out to test, update or replace each one. These updates and fixes can be easily accomplished through firmware over-the-air (FOTA) updates for your device's application software.

For example, let's say that you have developed a product that worked fine in your lab, but once your devices were deployed in areas with poor connectivity or higher than expected latency, they stopped working. You realize that your device's software needs to be updated to allow for longer response times, and you are very happy to be able to make this change using FOTA updates.


New Firmware Downloads to Each Device in the Field
Firmware Update Released to Cloud
Publish a New Firmware Change

Or perhaps you discover – after you've deployed hundreds of connected devices into remote areas – that there's an issue with your initial software design and the sensors have stopped sampling. The only practical way to fix this problem is to update the application firmware on the device using FOTA.

In situations such as these, using device FOTA updates is very preferable to physically visiting each device in the field and updating them one by one. All you'll need is a secure,

Continued on Page 10

encrypted communications channel and a new version of the application. You can then update the firmware in small batches or all at once.

Key Questions

- How do I want to set up the FOTA update process for my devices?
- Will I conduct a given update – in batches or all at once?

“Similarly, if something goes wrong with your deployed IoT devices, you’ll want to have the ability to remotely address the problem without sending a truck out to test, update or replace each one.”

Consideration 5

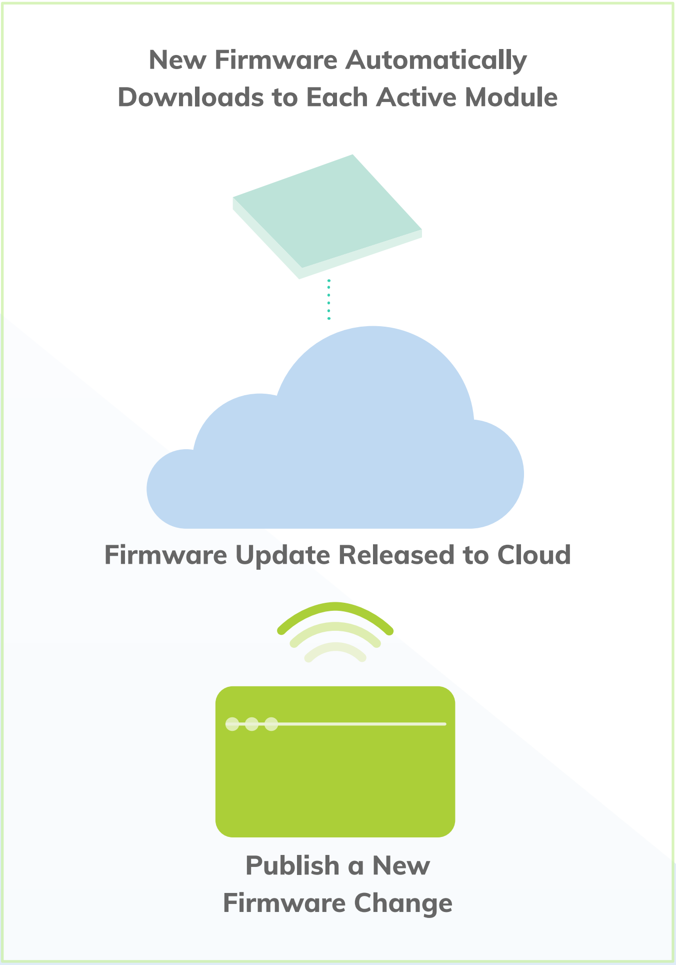
Allow Firmware Over-the-Air Updates for the Cellular Module

This is Not Optional! Your module's firmware is directly responsible for interacting with the cellular network, which is not under your control. And in order for a module to be approved and certified for use on a carrier's network, the module must allow for FOTA updates.

So if your network provider makes a network system or equipment change, you must enable that change to be made via FOTA updates to the module. Your device firmware must be designed to enable and facilitate FOTA updates to your module. If you fail to make this happen, your module may no longer be able to operate over that network.

This process is used to fix bugs or push out new features into the module. To enable this, you are required to build-in support for module FOTA. If you're using Skywire modems, NimbeLink will provide you with the guides and documentation that show you how to enable the module FOTA updates for the particular modem you are using.

The responsibility then falls on the device developer (you) to enable these updates and not get in the way of it happening. You must pick the most appropriate way to do this, either all at once to all your devices, or in a staged manner.


New Firmware Automatically Downloads to Each Active Module
Firmware Update Released to Cloud
Publish a New Firmware Change

Key Questions

- Will my FOTA updates to my modules be executed at one time across all devices, or will I stage these updates?
- What needs to be included in my device's firmware to support the FOTA method for my module?

"Your device firmware must be designed to enable and facilitate FOTA updates to your module."

Consideration 6

Designing Power Circuitry

Your IoT device will need power and LTE technology tends to demand power in short, high-current bursts, so how do we allow for the power to the device?

When you're designing your board and power circuitry, design-in the capability for accommodating short bursts of up to 2 amps of current. You also need to design for the very spiky nature of the power draw by including capacitance that can buffer the voltage fluctuations. The average current will be much less than 2 amps, but you must allow for spikes and we strongly recommend a maximum of 2 amps. All of the Skywire documents and schematics include this recommendation.

Another power consideration is the type of voltage regulator you will use. Because of its fast response time and high efficiency, a switching regulator is often preferable to a Low-Dropout (LDO) regulator on most devices. NimbeLink recommends using a switching regulator for cellular designs.

"When you're designing your board and power circuitry, design-in the capability for accommodating short bursts of up to 2 amps of current."

Key Questions

- Should I design my board and power circuitry from scratch, or use one of the suggested circuit designs from NimbeLink or the module manufacturer?
- Which voltage regulator will best meet our needs – a switching regulator or an LDO regulator?

Consideration 7

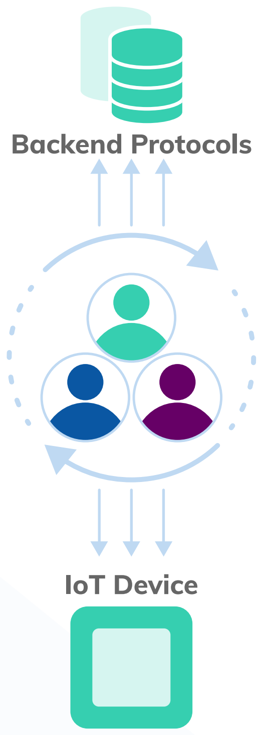
Simultaneously Developing Your Device and the Back-End Protocols

Many companies design an IoT device first, and then the backend platform to which the device will send its data. Others build the backend software platform first, and then a device that will feed it data. In both cases, doing one at a time often leads to significant delays in the deployment of an IoT solution.

So, if you want to get your IoT device to market quickly without delays, we recommend that you simultaneously develop both, and get the device designers and the software engineers working together from day one.

Your device will need to know what protocols are required by your platform, how to send the data, and what security protocols are required. And software folks will need to understand the use case and the device's specifications and capabilities to smoothly integrate it into the system architecture.

"...device designers and the software engineers working together from day one."



Certain cloud platforms require specific network features for security, say TLS1.2, an encryption protocol. If you are going to support TLS1.2, it might be a stretch for an 8-bit microcontroller to handle it. So, in that case, when you're picking a modem, make sure that you choose one that supports TLS1.2 so that you don't have to put that burden on your microcontroller. This is a good example of how simply knowing what cloud platform you'll be using and what its

Continued on Page 15

requirements are feeds directly into your device design. Or, perhaps, your device might also need a special chip to store the public encryption key.

Do your homework and begin by designing the device and the backend software platform in tandem. You'll accelerate your time to deployment and get the best possible solution for your use case.

Key Questions

- What type of protocols are supported by my chosen platform?
- Is it possible for my team to develop the backend and the device in tandem?

"Do your homework and begin by designing the device and the backend software platform in tandem. You'll accelerate your time to deployment and get the best possible solution for your use case."

Learn More

Selecting Your Antennas

- Skywire Hardware Developer’s Guide
- Skywire Hardware Design Checklist
- Can a Paperclip be an Antenna?
- Antennas and Connectivity 101
- Testing Signal Strength with Skywire

Logging Your AT Commands

- Skywire Software Developer’s Guide
- Using AT Commands and Skywire Development Kit
- Cellular Radio Software Requirements

Creating a Network Monitor

- Skywire Software Developer’s Guide
- Cellular Radio Software Requirements

Enabling FOTA Updates for the Device

- Skywire Software Developer’s Guide
- Understanding FOTA

Enabling FOTA Updates for the Module

- Skywire Software Developer’s Guide
- Understanding FOTA

Designing Power Circuitry

- Skywire Hardware Design Checklist
- Power of Skywire: Requirements and Design

Simultaneously Developing Your Device and the Back-End Protocols

- Getting Started with Cellular Training Module from Digi-Key
- Skywire Hardware Developer's Guide
- Skywire Hardware Design Checklist
- Skywire Software Developer's Guide
- How do I get started with a Skywire Embedded Modem?
- NimbeLink Let's Build! Create your own IoT Device
- Lets Build: Smart Mailbox Part 1
- Lets Build: Smart Mailbox Part 2

Conclusion

Our final recommendation is to read the data sheet for whatever cellular module or modem you intend to use. It will provide all you need to know about that particular radio and its requirements.

Ready to get started? Have more questions? **Contact our Product Support team** to discuss your application.

About Skywire modems

* **Skywire cellular modems** are designed to make cellular integration as fast and as easy as it can be. Exceptionally small, pin-compatible, and end-device certified, Skywire modems enable developers to quickly and reliably connect IoT products to cellular networks around the world. All Skywire modems are patented and are backed by industry leading documentation and expert-level product support.