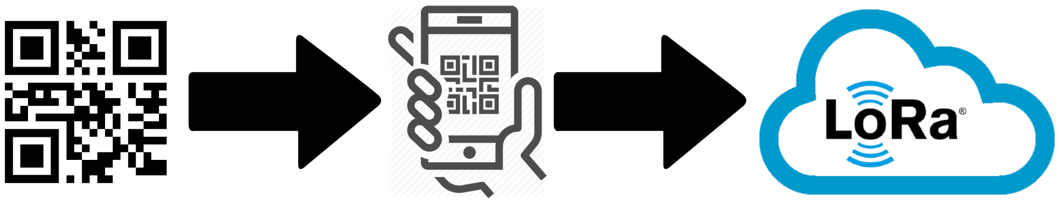
BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2022
http://excel.fit.vutbr.cz

Tool for Easy Device Configuration on LoRaWAN

Tomáš Moravčík*



Abstract
Modernization of the world's industry calls for automation. With businesses implementing various sensors, many variables arise that must be overseen. It is becoming common to need hundreds or even thousands of sensors spread over a large area. Nevertheless, setting up hundreds of such devices individually is far from automation. Utilizing Internet of Things (IoT) technology, this work solves the tedious process of manually implementing end devices into the LoRa network, specially developed for large-scale projects. The application will be PWA (Progressive Web App), designed to be reliable and installable on any device with a single codebase, ready to be used with significant convenience. Since sensors come with their unique IDs, represented both in human-readable form and QR code, the app can quickly scan these codes and create logs, all while being offline. Afterward, it processes this information into a registration request sent to the LoRa network server. This application would tremendously improve the handling time of each device from minutes to seconds.

Keywords: LoRa — PWA — Sensors — Optimization — Device configuration

Supplementary Material: Deployed Demo

*xmorav41@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Many businesses turn to automation to either save money on repetitive tasks or to modernize their informative system. This is especially true for large companies trying to micromanage their production, for example, quality checks, correct temperature or humidity levels, reserves quantity, et cetera. LoRa technology solves these issues on a large scale, regularly updating information with the minimum workforce needed. So LoRa supports hundreds of devices, and the range of transmission activity is tremendous compared to classic IoT devices in dense urban and rural areas. Possibilities with LoRa networks are compelling and should be thought of when expanding one's business.

Setting up LoRa Network requires quite a few steps. One of the most tedious steps is preparing and setting up numerous devices, presumably in a warehouse environment, with limited connection. The client would desire and expect swift installation, meaning we would need to trivialize this long and exhausting process done by a specialist into something that any layman could easily do. That's the goal of this application. It can easily scan each of these devices' QR code as input, by which it can generate a proper file that is used to register all scanned devices on the network. Furthermore, the app may log meta-information about the scanning process and create separate data used by the client's own database to overview all devices.

2. LoRa

LoRa is a radio frequency modulation technology for low-power, wide area networks (LPWAN). The name

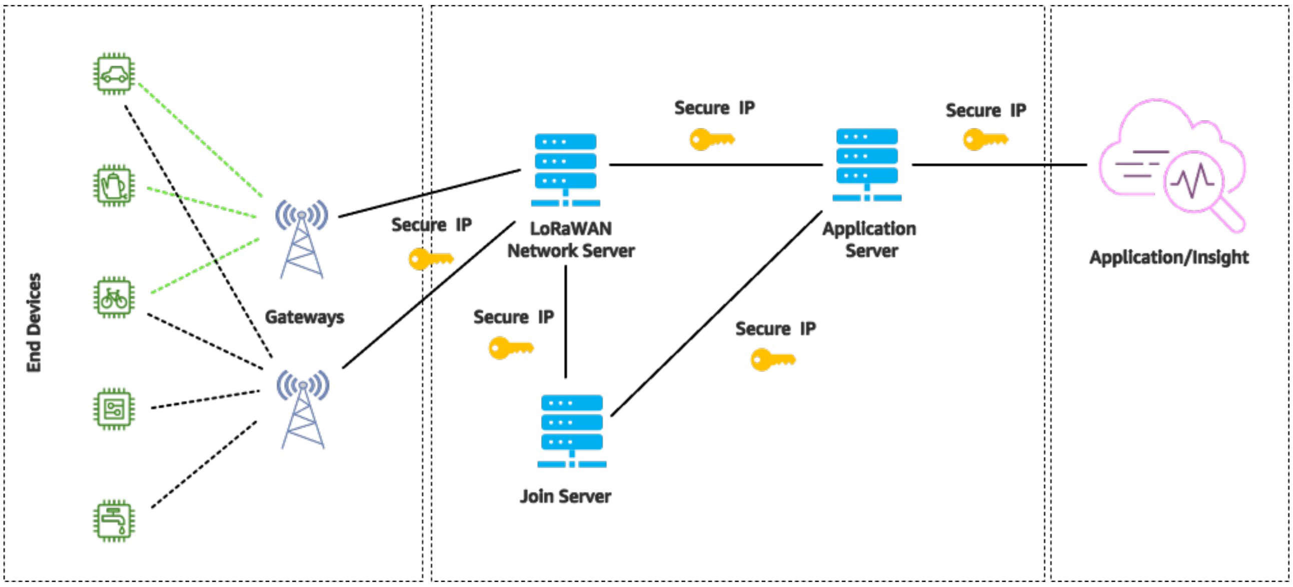


Figure 1. LoRaWAN Network Architecture.

refers to long-range data links that are possible in LoRa. Semtech created it to standardize LPWANs. Thanks to LoRa, communications are guaranteed up to five kilometers in urban and fifteen kilometers in rural areas. Another huge factor is the low-power requirement, which allows devices to last up to ten years on a battery. The network deployed in star topology allows applications to collect data from many devices in a large area.

Compared to traditional cellular technology, LoRa trades high data rates for longer battery life, lower cost, and high message capacity. LoRaWAN network capacity can support millions of messages, depending on how many gateways are installed. A single eight-channel gateway will support a few hundred thousand messages over 24 hours[1].

2.1 LoRaWAN Device Activation

To transmit messages to and from the end device, it must be registered with a network. This process is called activation, and there are two possible methods:

- Over-The-Air-Activation (OTAA) - Secure and recommended method. The device performs a join procedure with the network and is assigned a dynamic address and derived security keys.
- Activation By Personalization (ABP) - The device address and security keys are hard-coded. Devices also cannot switch network providers without manually changing keys in the device.

This tool utilizes the former.

2.2 OTAA in LoRaWAN 1.0.x

The join procedure between the end device and the Network Server requires two MAC messages:

- Join-request: from end device to the Network Server
- Join-accept: from Network Server to the end device

2.3 Identity Server

The Identity Server provides the registries that store entities such as applications with their end devices, gateways, users, organizations, OAuth clients, and authentication providers. It also manages access control through memberships and API keys[2].

2.4 Join Server

The Join Server (JS) is responsible for the over-the-air activation (OTAA) process of end devices wanted to be added to the network. The server performs mutual authentication with the end device and notifies the LoRaWAN Network server to which Application Server should end-device connect. Join servers are uniquely identified by a 64-bit identifier, AppEUI / JoinEUI. End devices derive session keys based on their identifiers like DevEUI and JoinEUI so that these keys won't be transferred over the air. JS contains this information about each end device in its control:

- DevEUI (End-device serial unique identifier)
- Root Keys:
  - AppKey (Application encryption key)
  - NwkKey (Network encryption key)
- Application Server Identifier
- End Device Service Profile

2.5 Application Server

The Application Server handles the LoRaWAN application layer, including uplink data decryption and decoding, downlink queuing, and downlink data encoding and encryption[3]. It is responsible for handling, managing, and interpreting data from sensors. This also feeds data to the dashboard UI.

2.6 Network Server

The Network Server handles the LoRaWAN network layer, including MAC commands, regional parameters, and adaptive data rate[4].

3. Utilized Technologies

During the research on how to conduct this work, two main issues were considered. First, the application should be cross-platform. Secondly, finding a way to set up the device in the network with the least effort possible. The concept of Progressive Web Applications solves the first problem, while the second is made possible by finding LoRaWAN Network Server, which supports HTTP methods. Lastly, it was all needed to put together with a meaningful web application framework.

3.1 Progressive Web Applications

Application's prominent feature is usability between devices regardless of their platform. PWA stands for Progressive Web Apps that are capable, reliable, and installable on any device with a single codebase. PWA are built with modern APIs, which offer the experience of platform-specific apps. Whether the client uses mobile Android, iOS, or desktop browsers, the app will ensure a smooth, responsive, secure experience and offline functionality. Installed PWA run in a standalone window instead of the browser tab. They can be launched from the home screen, and it's possible to search for them and switch them between apps.

The most notable platform with huge results is Twitter, with 80% of mobile users. After switching to PWA in 2017, they saw a 65% increase in pages per session and a 75% increase in sent tweets. By other statistics, Progressive Web Apps register 50% higher user engagement compared to native apps, up to 10 times faster loading speed, three times less development cost, and one third cheaper maintenance. Another very nice benefit is a small size, often under 1 MB. With all these advantages, the bounce rate of PWA is around 43%. [5]

The main difference between PWA applications from regular ones is including Manifest file and Service worker.

3.1.1 Manifest

Manifest is a JSON file that informs the browser about PWA capability and how it should behave when installed. The file usually includes the app name, icons, and starting URL.

3.1.2 Service worker

Service worker is essentially a JavaScript file that runs separately from the main browser thread, intercepting network requests, caching or retrieving resources from the cache, and delivering push messages. This makes the associated service worker independent of the app. The service worker lets us control how network requests from the page are handled. Creating the service worker consists of three steps:

- Registration
- Installation
- Activation

In the main JavaScript file, registration tells the browser where the service worker is located. Then, the installation can be attempted if a service worker is considered new by the browser. Afterward, the service worker proceeds to the activation stage and updates the page's cache.

3.2 The Things Stack Cloud

The role of the LoRaWAN Network Server is filled by The Things Stack (TTS) Cloud. It is hosted software offered by The Things Industries, which specializes in products and services for IoT developers and businesses. TTS provides a way to manage devices, gateways, data, and most importantly, HTTP API. With a free subscription, it was possible to test and evaluate the benefits of TTS Cloud. Furthermore, another reason for choosing this particular provider is familiarity and usage inside the Logimic.

3.3 Angular framework

Angular is a development platform built on TypeScript (a strongly typed programming language that builds on JavaScript). Angular follows component-based architecture for the development of scalable web applications. Components are building blocks of UI that are independent and reusable. It consists of a template, class file, and styling file. Components follow tree structure; for example, the parent scanner component consists of child camera capturing and button components. Upon a change in some components that would be informative to the user, only that component rerenders instead of the whole document. This saves loading time and improves the user's experience.

Modules in Angular are a group of components, directives, pipes, and services that serve common functionality, for example, the HTTP module used to make HTTP calls. Each Angular application must have a root module. The first file that is executed is `main.ts`. It loads everything and controls the application's startup. `app.module.ts` file is referred here to list all components that should be known to Angular before analyzing and rendering the `index.html` file, giving the user a fully prepared page.

4. Requirements

In order to register the end device in TTS Cloud, the user needs to manually input 16 characters long DevEUI and AppEUI and 32 characters long AppKey, as well as other device settings. This process becomes very time-consuming, considering dozens or hundreds of devices, and is more prone to human error. The tool circumvents this interface and trivializes it into a simple scan and send process.

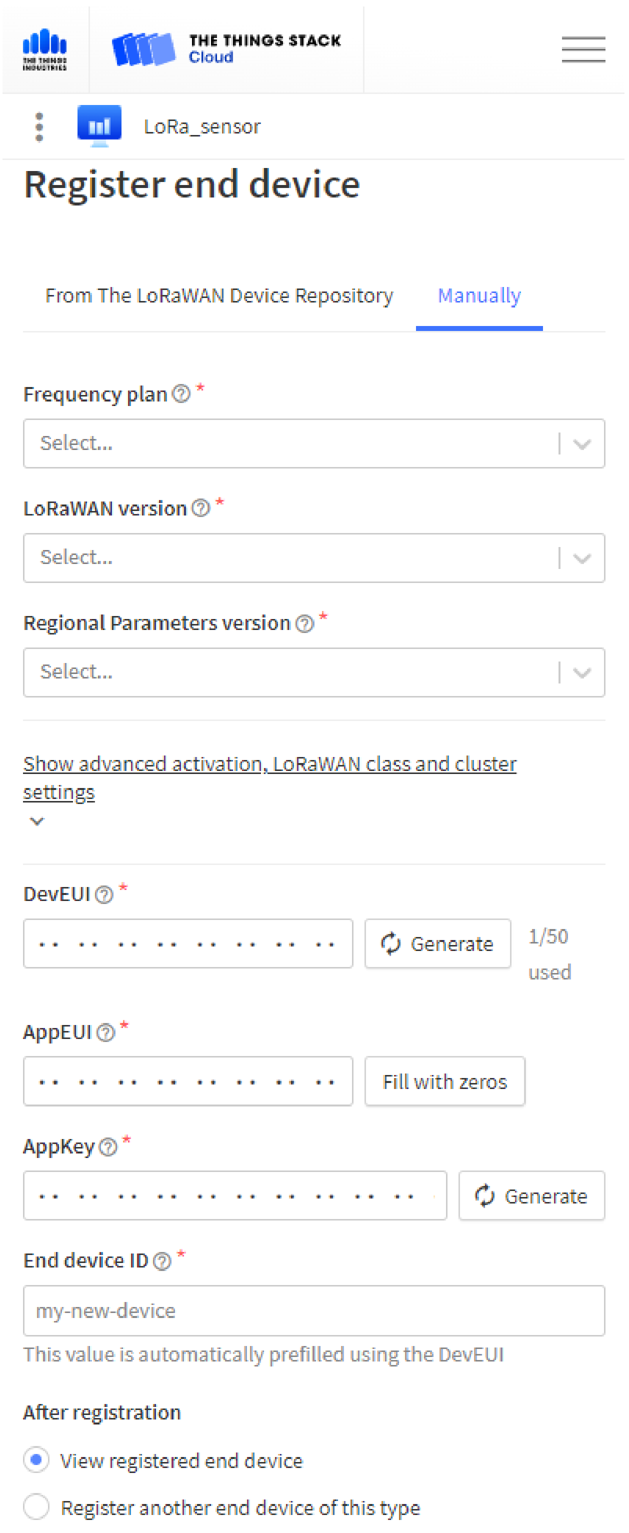
4.1 Collaboration with Logimic

The application, which I have chosen as my bachelor thesis, is developed in cooperation with Logimic. Logimic is a company specializing in providing IoT applications for industries with a focus on wireless devices control, web applications, and software services. With their help, the application is tested on real LoRa devices. The main objective of this tool is to increase efficiency and improve customer service.

5. Design

Usually, the device comes with a single QR code containing DevEUI. We can easily derive AppEUI from this information. The AppKey is given based on which application will operate the device. Lastly, the tool derives the End device ID, also using the DevEUI.

When working with the tool, the user logs in with the company's credentials and may pick the appropriate device model and cloud application for the device. Now the user will choose to scan the device via camera or manually input information in case the code is obstructed. After successful scanning, to register devices to The Things Stack Cloud, four different servers must be contacted. These are the aforementioned Identity Server, Join Server, Application Server, and Network Server. Each one is given a specific JSON file containing device IDs and configurations. Communication is implemented via an HTTP request to the cloud. In detail, the application sends one `POST` request to the Identity Server and then, in total, three `PUT` requests to the rest of the servers. Cloud responds either with a successful device registry or error. With the PWA implementation, the app can be installed on any device and may be used in offline mode to scan devices. When connection to the internet is restored, the user


THE THINGS STACK
Cloud

LoRa_sensor

Register end device

From The LoRaWAN Device Repository | Manually

Frequency plan * : Select...

LoRaWAN version * : Select...

Regional Parameters version * : Select...

Show advanced activation, LoRaWAN class and cluster settings

DevEUI * : .. .. | Generate | 1/50 used

AppEUI * : .. .. | Fill with zeros

AppKey * : .. .. | Generate

End device ID * : my-new-device

This value is automatically prefilled using the DevEUI

After registration

[✓ View registered end device]

Register another end device of this type

Figure 2. The Things Stack Cloud's device register interface.

may register those devices to the TTS Cloud. This process replaces the current manual input required to register a device.

6. Conclusion

The primary purpose of this work is to address the slow and manual process of integrating a large number of

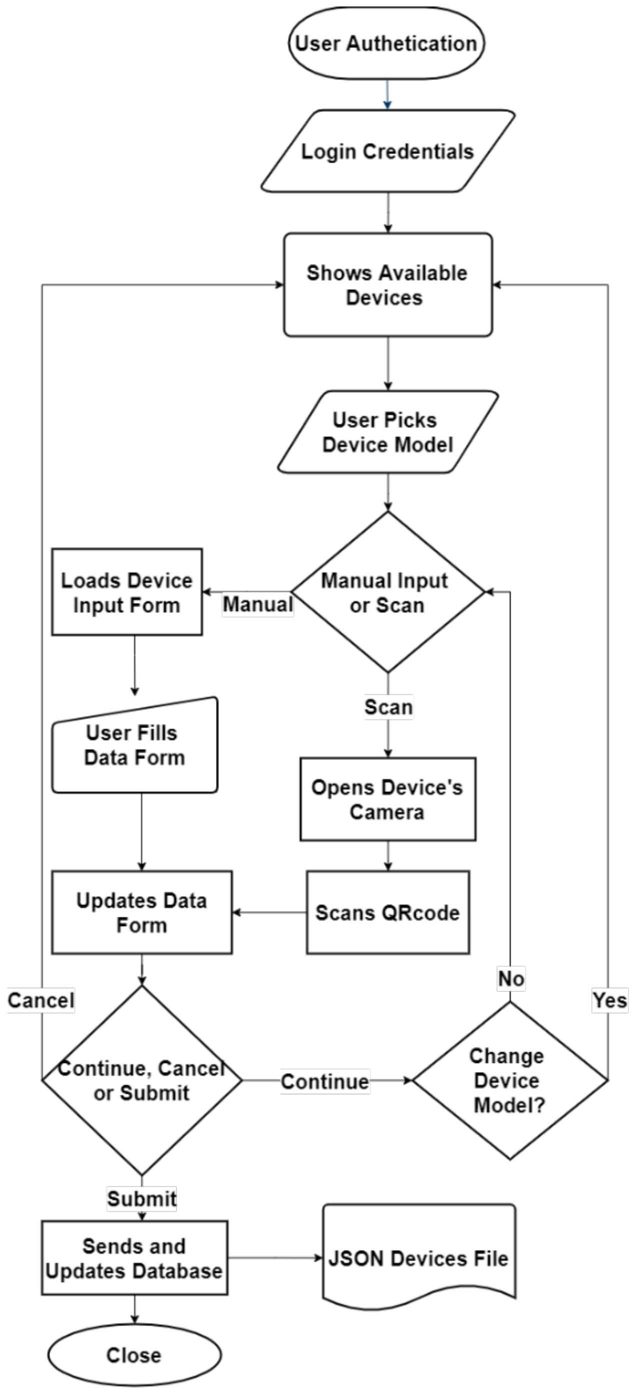


Figure 3. Workflow application.

devices into LoRaWAN. This paper provided a base to understand LoRa technology and the initial issue of client service when installing a new system. The application helps to save the time it takes to set it up and offers additional features for logging data. I believe this work will be improved in the future to support new devices with new configuration needs and clients' input. The work is done in collaboration with **Logimic** company with a signed NDA.

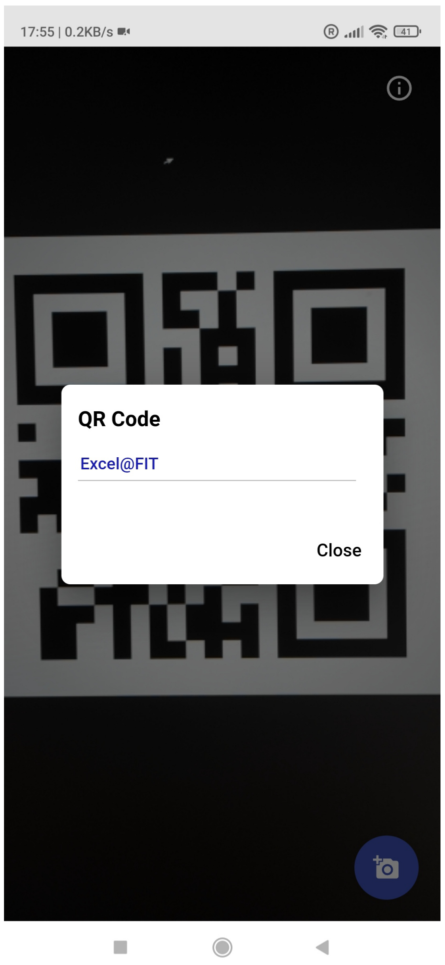


Figure 4. Scanner's interface.

7. Acknowledgment

I would like to thank my supervisors, Jiří Hynek and Michal Valný, for mentoring and helping me with the project.

References

[1] What are lora® and lorawan®? https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan.

[2] Identity server. https://www.thethingsindustries.com/docs/reference/components/identity-server/. Accessed: 2022-04-22.

[3] Application server. https://www.thethingsindustries.com/docs/reference/components/application-server/. Accessed: 2022-04-22.

[4] Network server. https://www.thethingsindustries.com/docs/reference/components/network-server/. Accessed: 2022-04-22.

[5] Pwa statistics. https://www.pwastats.com/. Accessed: 2022-03-26.