netidee PROJEKTE

# LoRaBridge 2

## Weitreichende Automatisierungen

Zwischenbericht | Call 18 | Projekt ID 6693

# Inhalt

# 1 Einleitung

Das Hauptziel des Projektes LoRaBridge 2 ist es, die Konfiguration und Überwachung von Heimautomatisierungen an entfernten Standorten zu ermöglichen. Da die meisten kommerziellen Automatisierungslösungen für Privatpersonen eine Internetverbindung erfordern, wird selbst die Einrichtung einfachster Automatisierungen umständlich. Wir wollen ein alternatives open source Home Automation Gateway anbieten, bei dem die Internetverbindung durch eine LoRaWAN (Long-Range Wireless Standard) Verbindung ersetzt wird. Auf diese Weise können Benutzer*innen Automatisierungen z. B. auf einem Heimlaptop vorbereiten und über ein öffentliches oder privates LoRaWAN-Netzwerk zum Gateway übertragen. Da es sich bei diesem Projekt um ein Nachfolgeprojekt des Netidee-Projekts LoRaBridge handelt, bei dem der Schwerpunkt auf der Reichweitenerweiterung für drahtlose Sensoren lag, konnten wir LoRaBridge 2 starten, indem wir einige der Funktionalitäten wiederverwendeten, die wir bereits im Jahr 2022 implementiert hatten.

Wir konnten das Projekt wie geplant pünktlich starten. Zu den ersten Aufgaben gehörte die Auswahl geeigneter Hardwarekomponenten und Open-Source-Softwarebibliotheken/-Frameworks. Wir haben diesbezüglich einen besonderen Fokus auf eine Basis gelegt, die sowohl Kompatibilität, Verfügbarkeit als auch Modularität bietet. Zum Beispiel haben wir uns entschieden, die für LoRaWAN erforderliche Hardware/Software auf die neuesten Entwicklungsmodule/Bibliotheken zu aktualisieren. Auf diese Weise kann LoRaBridge 2 von mehreren LoRaWAN-Modulen betrieben werden. Weiters funktioniert die LoRaWAN-Verbindung auch außerhalb der EU, da mehrere Frequenzkonfigurationen unterstützt werden. Auf der Softwareseite bevorzugten wir beliebte Frameworks, die aktiv in der Entwicklung sind und eine starke Open-Source-Community haben. Beispielsweise haben wir das Node-RED Framework ausgewählt, eine der beliebtesten Optionen für die Programmierung von Automatisierungen.

Eines der entscheidenden technischen Hindernisse, die es zu lösen galt, war die Übertragung der Automatisierungsregeln über eine LoRaWAN-Downlink-Verbindung. Da die Downlink-Datenrate sehr begrenzt ist, würde das Senden einer einfachen unkomprimierten Node-RED Datei mehrere Tage dauern. Unsere Lösung, um die Konfigurationslatenz zu reduzieren, bestand darin, das sogenannte „Dictionary based compression“ Komprimierungsverfahren zu adaptieren, welches wir bereits im LoRaBridge-Projekt für andere Daten erfolgreich eingesetzt haben. Ein großer Teil der Arbeitsleistung wurde für diesen Bereich aufgewandt, da wir sowohl Kompressionsregeln als auch eine Software entwickelt haben, die sich um die Dekomprimierung und Generierung von Node-RED-Flows kümmert. Diese Kombination stellte sich als erstes Projekt-Highlight heraus, da wir mit unseren Kompressionsregeln ein Kompressionsverhältnis von 300 erreichen. Das bedeutet, dass die Übertragungszeit für eine Automatisierungsregel meist wenige Minuten (bis zu einigen Stunden für eine Worst-Case-Verbindung) statt einiger Tage dauert.

# 2 Status der Arbeitspakete

## 2.1 Arbeitspaket 1 – Detailplanung und Formales am Projektstart

Das Projekt wurde mit der Erstellung eines detaillierten Zeit- und Arbeitsplans, der pünktlich an Netidee geliefert wurde, angefangen. Der Plan besteht kurz gefasst aus drei Hauptteilen: 1) Planung und Forschung (AP2, AP3), 2) Prototyp-Implementierung (AP4, AP6) und 3) Validierung (AP5). In der ersten Phase haben wir uns damit beschäftigt, die beste, verfügbare Software und Hardware auszuwählen. Darüber hinaus werden wir vor dem Prototyping die Konfiguration der Heimautomatisierung über LoRaWAN festlegen, die Definitionen für unterstützte Automatisierungsgeräte und -algorithmen, Komprimierungsregeln, das LoRaWAN-Downlink-Protokoll für die Datenübertragung sowie die verwendeten Datenstrukturen enthält. Die eigentliche Implementierungsarbeit findet in der zweiten Phase statt, die hauptsächlich aus der Software-/Firmware-Entwicklung besteht. Die dritte Phase umfasst die Validierung der Konfigurationsprogrammierung für die Hausautomation über LoRaWAN. Der Fokus liegt hier darauf, sicherzustellen, dass keine fehlerhaften Konfigurationen z.B. aufgrund fehlender LoRaWAN-Pakete im Home Automation Gateway landen.

Gleich zu Beginn des Projekts haben wir eine GitHub-Organisation für das Projekt zur Versionskontrolle und für das Issue-Tracking vorbereitet. Für die Aufgabenverfolgung/das Brainstorming verwendeten wir weiterhin das Trello-Board, was sich bereits während des ursprünglichen LoRaBridge-Projekts als vorteilhaft erwiesen hat. Zu Beginn des Projekts wurde auch der erste Netidee-Blogbeitrag erstellt.

## 2.2 Arbeitspaket 2 – Recherche Software/Hardware/Schnittstellen

Da unser Long-Range Automatisierungskonzept 3rd-Party-Frameworks für z.B. die ZigBee Schnittstelle und LoRaWAN-Anbindung erfordert, war es wichtig, die neuesten verfügbaren open source Optionen zu nutzen. Bei der Auswahl der geeigneten Hardware und Software waren unsere Hauptkriterien zum einen die Verfügbarkeit (z. B. Unterstützung für mehrere Funkmodule) und zum anderen die aktive Wartung/Community-Support.
Auf der Hardwareseite haben wir weiterhin die Raspberry PI 4B+ Plattform verwendet, da sie ein gutes Gleichgewicht zwischen verfügbaren Rechenressourcen, Energieverbrauch und Anschaffungspreis bietet. Daher haben wir das Raspberry PI LoRa-Modul von LoRaBridge 1 durch ein ESP32 LoRaWAN Entwicklungsboard ersetzt. Wie sich herausstellte, ist die LoRaWAN-Stack-Unterstützung für das Erstgenannte etwas veraltet, was unseren Fokus auf alternative LoRaWAN Bibliotheken lenkte. Diese Designentscheidung bringt einige optionale Vorteile mit sich, wie z. B. die Unabhängigkeit des Raspberry PI als Automatisierungsgateway. Zweitens könnte man die WLAN-Schnittstelle des ESP32 benutzen, um Daten vom Raspberry PI zum LoRaWAN-Modul zu leiten, um

die LoRaWAN-Antenne flexibler zu platzieren. Dieses optionale Designziel kann die LoRaWAN-Reichweite in Anwendungsfällen verbessern, in denen die LoRaWAN-Verbindung an ihre Grenzen stößt. Der Rest der Hardware ähnelt jener, die im ursprünglichen LoRaBridge-Projekt verwendet wurde. Das Raspberry PI LoRaWAN-Gateway-Modul und der TI CC2652-basierte ZigBee-Wireless-Adapter werden in LoRaBridge 2 weiter zum Einsatz kommen.

Auf der Softwareseite wurde die Unterstützung von LoRaWAN-Modulen auf das LMIC-Node-Repository aktualisiert, bei dem es sich um einen LMIC-basierten LoRaWAN-Stack handelt, der auf die ESP32-Mikrocontrollerfamilie ausgerichtet ist. Somit kann die bereits implementierte Automatisierungskonfiguration über LoRaWAN mit über 10 verschiedenen LoRaWAN-Entwicklungsmodulen realisiert werden. Das LMIC-Node-Repository bietet auch Unterstützung für andere Frequenzbänder, z. B. in den USA oder Indien. Da das Repository den LoRaWAN-Standard 1.0.3 implementiert, können die Netzwerk-Timesync Anfragen verwendet werden, um die Zeit des Heimautomatisierungs-Gateways über LoRaWAN zu aktualisieren. Als Automatisierungsframework haben wir das beliebte Node-RED-Framework gewählt, das auf einer visuellen Flow-basierten Programmierung basiert. Als eines der beliebtesten open source Tools zur Implementierung von Automatisierungen mit Unterstützung für das Self-Hosting auf einem Raspberry PI, ist es eine gute Basis für das LoRaBridge Automation Gateway.

## 2.3 Arbeitspaket 3 – Automatisierungskonzept

Das Hauptziel des AP3 war es, ein Konzept für die Heimautomatisierung zu entwickeln, das die Einrichtung und Konfiguration über einen LoRaWAN-Downlink ermöglicht. Schließlich haben wir ein vereinfachtes Abstraktionsmodell entwickelt, das aus verschiedenen Aktionen, Knoten (Nodes) und Verbindungen zwischen Knoten besteht. Da es in unserem Modell nur eine begrenzte Anzahl an möglichen Geräten und Automatisierungen gibt, konnten wir die Informationen, die zur Darstellung von z.B. Knoten/Aktionen benötigt werden, auf ein absolutes Minimum reduzieren. Um eine Vorstellung davon zu bekommen, wie diese Komprimierung funktioniert, benötigt in unserem Modell beispielsweise eine Aktion zum Hinzufügen einer Automatisierungsregel nur vier Bytes. Im Vergleich dazu würde eine äquivalente Node-RED-Aktion mit JSON-Objekten einige hundert Bytes benötigen. Daraus ergibt sich ein Kompressionsverhältnis von ca. 300, was eine Automatisierungskonfiguration über LoRaWAN ermöglicht.

Die folgenden Aktionen wurden umgesetzt:

- *Knoten hinzufügen*
- *Knoten entfernen*
- *Gerät hinzufügen*
- *Aktualisierung der Parameter*
- *Knoten verbinden*
- *Knoten trennen*
- *Hinzufügen eines Flows**

- *Flow aktivieren*
- *Flow deaktivieren*
- *Durchfluss entfernen*
- *Ablauf des Uploads*
- *Zeit-Sync-Anfrage*
- Ablauf abgeschlossen

*(*Ein Flow ist eine Sammlung von mehreren zusammenverbundenen Knoten)*

Die folgenden Knoten sind unterstützt:

- Binäres Gerät (z. B. ein Smart Plug)
- Binärer Sensor (z.B. ein Bewegungssensor)
- Numerischer Sensor (z. B. ein Temperatursensor)
- Logisches UND
- Logisches ODER
- Zeitschaltuhr
- Hysterese
- Countdown-Schalter
- (Benutzerdefinierter Knoten)

Die Automatisierungsknoten, wie z.B. die Zeitschaltuhr, wurden in JavaScript implementiert, wie es in Node-RED unterstützt wird. Der Workflow zur Implementierung von Automatisierungsknoten beginnt mit der Definition einer Node-RED JSON-Vorlage. Anschließend sollen die Knoteneingänge, -ausgänge sowie konfigurierbare Parameter in das Template eingefügt werden. Mit einem solchen Workflow können Benutzer*innen benutzerdefinierte Automatisierungsknoten implementieren. Die oben genannten Knoten und Aktionen sollen die Anforderungen einfacher Use-Cases erfüllen, auf die LoRaBridge 2 abzielt.

## 2.4 Arbeitspaket 4 – Proof-of-Concept Prototyp

Der erste Prototyp des in AP3 konzipierten Hausautomationskonzepts wurde erfolgreich umgesetzt. Die zentralen Komponenten, die wir verwendet haben, waren Node-RED, ChirpStack (von LoRaBridge 1), die LoRa-Node-basierte Firmware für das LoRaWAN-Modul und die in Python geschriebene Automatisierungsmanager-Software. Die intensivste Entwicklungsphase umfasste die Programmierung des Automatisierungsmanagers und der Firmware für die LoRaWAN-Downlink-Kommunikation. Im Folgenden finden Sie einen kurzen Überblick über die implementierten Softwarekomponenten.

*Automatisierungsmanager*: Die komprimierten Automatisierungsbefehle werden aus Redis Datenbank bezogen und anschließend geparst. Der Manager prüft dann auf fehlerhafte/nicht unterstützte Befehle/Aktionen und führt die gewünschte Funktion aus. Sobald eine Automatisierungseinrichtung abgeschlossen ist, führt der Manager die Dekomprimierung der definierten Aktionen/Knoten in einen Node-RED-Flow durch. Schnittstellen zu Zigbee2MQTT und zu Node-RED werden über MQTT bzw. REST realisiert. Die MQTT-Schnittstelle wird verwendet, um ZigBee-Geräte von Benutzer*innen den Geräteindizes des Automatisierungsmanagers zuzuordnen. Die Node-RED REST-Schnittstelle wird verwendet, um den generierten Node-RED-Flow auf Node-RED hochzuladen.

*LoRaWAN-Modul:* Die Firmware für das ESP32-basierte LoRaWAN-Modul übernimmt zwei Aufgaben. Zunächst werden die Sensor-/Gerätedaten vom Bridge-Gerätemanager über eine serielle USB-Schnittstelle abgerufen und über eine LoRaWAN-Verbindung zu ChirpStack geschickt. Zweitens werden komprimierte Automatisierungskonfigurationspakete über LoRaWAN-Downlink (als Teil von MAC-Befehlen und ACK-Paketen) an das Modul übertragen. Falls keine Gerätedaten für die Uplink-Übertragung vorgesehen sind, sendet die Firmware ein "Heartbeat"-Paket, damit eine periodische Downlink-Kommunikation stattfinden kann.

## 2.5 Arbeitspaket 5 – Validierung und Tests

Erste Validierungstests wurden für den Automation Manager vorbereitet, der viele zu verifizierende Funktionen in sich trägt. Zunächst wurden Testskripte geschrieben, die Automatisierungskonfigurationsbefehle direkt in die Redis-Datenbank der Automation Hub Unit pushen. Auf diese Weise wurde die Fehlersuche/-behebung beschleunigt, da die Latenz durch das LoRaWAN vermieden werden konnte. Sobald wir die minimal funktionsfähige Version des Automatisierungsmanagers erreicht hatten, wurde ein einfaches LoRaWAN-Downlink-Scheduler-Skript geschrieben, um die Automatisierungskonfiguration über LoRaWAN zu testen.

Mit der Planung von Automatisierungsbefehlen haben wir die Konfiguration von zwei einfachen Automatisierungen getestet: Lichtsteuerung per Zeitschaltuhr und bewegungssensorgesteuerte Leuchten. Den Validierungsergebnissen zufolge können einfache Automatisierungen über LoRaWAN in etwa 1-60 Minuten konfiguriert werden, abhängig von der Qualität der LoRaWAN-Verbindung.

Alles in allem bestätigten die positiven Ergebnisse die Funktionalität unseres Systems, was definitiv ein Highlight des ersten Projekthalbjahres war.

### 2.6 Arbeitspaket 6 – Benutzeroberfläche

Der AP6 wird erst ab Anfang 2. Projekthalbjahr gestartet und daher werden die dazugehörigen Ergebnisse in Endbericht veröffentlicht.

### 2.7 Arbeitspaket 7 – Projektmanagement und Kommunikation

Vor dem offiziellen Projektstart erstellten wir den Projektarbeits- und Zeitplan, der pünktlich bei Netidee eintraf. Im Frühjahr 2024 haben wir periodisch Status-Update-Meetings abgehalten und zusätzlich einen halbtägigen Workshop organisiert, in dem das endgültige Systemmodell von AP3 definiert wurde.

## 3 Umsetzung Förderauflagen

Für das Projekt LoRaBridge 2 wurden keine besonderen Anforderungen oder Vereinbarungen zur Finanzierung getroffen.

## 4 Zusammenfassung Planaktualisierung

*Alle Anpassungen des Plan-Excels kurz zusammengefasst*

Der Hauptunterschied zwischen dem Projektplan und dem gemeldeten Fortschritt betrifft die Personenstunden. Für AP2-AP5 im Q1-Q2/2024 wurden die Aufgaben von H. Ruotsalainen ausgeführt. Daher sollen die in AP6 (Q3-Q4/2024) geplanten Aufgaben überwiegend von T. Dam erledigt werden. Der Grund für diese Planänderung ist, dass T. Dam in Q1/Q2 2024 in anderen Forschungsprojekten beschäftigt war. Da H. Ruotsalainen jedoch über ein ausgeprägtes Know-how in den Bereichen LoRaWAN, Heimautomatisierung, Systemdesign und Prototyping verfügt, waren weniger Arbeitsstunden erforderlich, um die geplanten Aufgaben zu erfüllen. Daher mussten wir auch keine Kompromisse bei den gesetzten Projektzielen eingehen und das Projekt kann wie geplant durchgeführt werden. Eine leichte Abweichung vom ursprünglichen Zeitplan betrifft den

Starttermin für AP6. Das AP6 soll im Juli 2024 beginnen, was sich ebenfalls nicht auf die Projektergebnisse auswirken soll.

# 5 Öffentlichkeitsarbeit/ Vernetzung

*Beschreibung der bereits erfolgten Öffentlichkeitsarbeit oder Vernetzung, bzw. Beschreibung des Plans künftiger Aktivitäten*

Im Frühjahr 2024 haben wir über unsere Fortschritte im Blog der Netidee Projektseite berichtet. Darüber hinaus haben wir im März das ursprüngliche LoRaBridge-Projekt in einem Hackrnews-Beitrag "Show HN" geteilt, der eine gute Sichtbarkeit erlangte und schließlich auf die Titelseite gelangte. Dies brachte uns konstruktives Feedback, das auch für LoRaBridge 2 relevant ist.

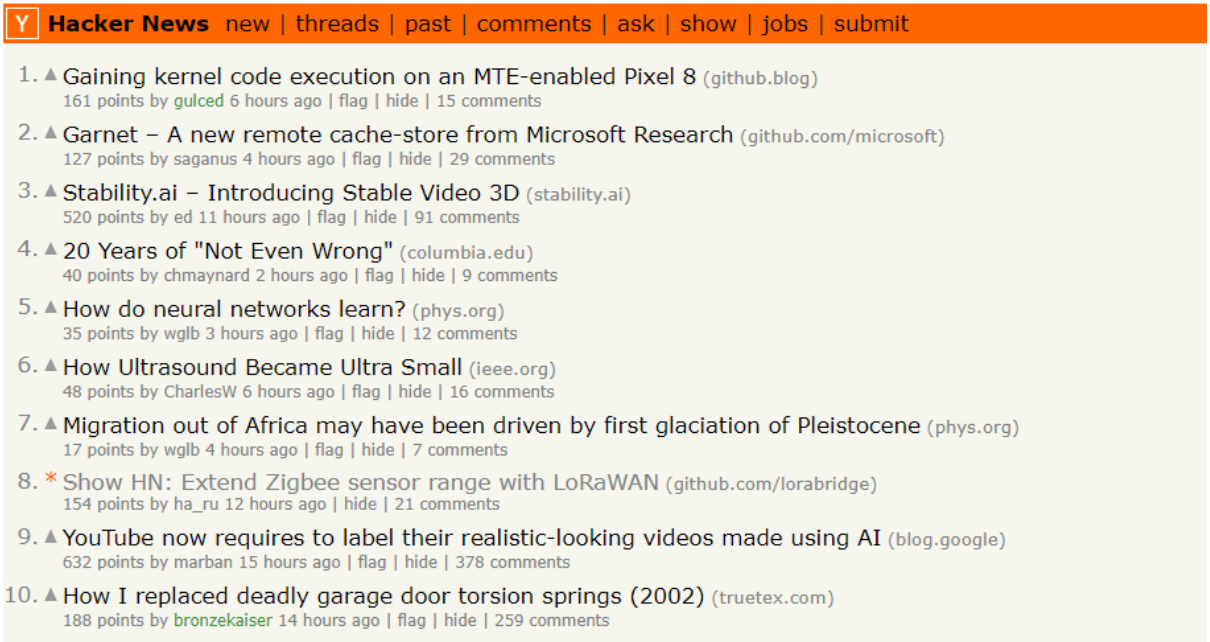


*Abbildung 1- LoRaBridge in Hackrnews Frontpage (8. Show HN...)*

Zu den bevorstehenden Verbreitungsplänen gehören:

- Weitere Beiträge in sozialen Medien wie Hackrnews/Reddit und GitHub-Foren
- Erstellung eines Discord-Servers für LoRaBridge 2, Automatisierung, Freigabe und technische Diskussion
- Fachvorträge auf Konferenzen/Veranstaltungsorten

# 6 Eigene Projektwebsite

*Wird zusätzlich zur netidee-Projektwebsite noch eine eigene Website betrieben, so ist hier die Adresse anzugeben.*

Wir haben nicht vor, eine www-Domain für LoRaBridge 2 zu registrieren.