

# WALDBRAND- FRÜHERKENNUNG

---

mit LoRaMesh



# Waldbrandfrüherkennung mit LoRaMesh

## Ausgangssituation

Durch den Klimawandel steigt die Waldbrandgefahr signifikant an, da längere Trockenperioden, höhere Temperaturen und häufigeres „Feuerwetter“ (Hitze, geringe Luftfeuchtigkeit, Wind) die Austrocknung von Vegetation beschleunigen.

Obwohl menschliche Aktivitäten wie Lagerfeuer oder Brandstiftung als Hauptursachen gelten, bleibt ein Großteil der Auslöser ungeklärt - in Deutschland sind über 50% der Waldbrandursachen unbekannt. Bisherige Forschungen vernachlässigten unseres Erachtens kritische Faktoren wie:

- Bodenfeuchte-Variabilität (entscheidend für Brandausbreitung) sowie
- Schadstoffemissionen in Bodennähe.

Insbesondere VOC (flüchtige organische Verbindungen) und extrem hohe CO-Konzentrationen wurden trotz ihrer ökologischen Risiken kaum systematisch erfasst.

## Lösungsansatz

Unser Konzept integriert erstmals Feuerwinde (durch Thermik verstärkte Winde), Bodentemperaturmodelle (basierend auf thermischer Diffusivität) und Echtzeit-Messungen von Gasen. Die Daten werden vor Ort gesammelt, ausgewertet und per Funk übertragen.

Dieses System ermöglicht eine Früherkennung innerhalb wenige Minuten nach Brandausbruch, selbst in schwer zugänglichen Waldgebieten.

## Herausforderung

In dichtbewachsenen Wäldern ist eine direkte Datenübertragung per Funk nur mit hohem technischen und energetischem Aufwand realisierbar. Die Energiebereitstellung durch erneuerbare Energie, wie PV-Module und Ladezellen ist durch die bodennahe Montage erschwert bzw. nur in Waldlichtungen möglich. Wird ein engmaschiges Monitoring angestrebt, sind der Energieverbrauch der Sensoren und die Funkübertragung entscheidende Faktoren, damit die Funktionsdauer über Jahre ohne Wartungsintervalle gewährleistet werden kann.

## Executive Summary

In unserem Projekt haben wir uns zunächst auf die primären Rahmenbedingungen, wie

- **Funkübertragung** und
- **Energieverbrauch**

fokussiert, da diese entscheidend für den Erfolg der Waldbrandfrüherkennung sind und einen möglichen Ansatz für weitere Projekte im Umwelt-, Klima- und Katastrophenschutz (wie zum Beispiel bei Überschwemmungen, ausgelöst durch lokalen Starkregen) sein können.

In unseren Lösungsansätzen können wir den Eigenenergieverbrauch durch geeignete Bauelemente und zeitliche Funktionsslots stark einschränken. Ein Betrieb des Systems mit einer Langzeitbatterie (19Ah) ist, unseren Messungen und Berechnungen nach, für die Dauer von bis zu 10 Jahren ohne Wartung denkbar.

Die bodennahe Datenübertragung aus dichtbewachsenen Gebieten ist über ein LoRaFunkMesh realisierbar. Dabei werden die Informationen vom Sensor über mehrere Hubs bis an das Ziel (Zugang zum Internet) übertragen. Das Meshnetzwerk ist in zufälligen aber synchronen Zeitslots aktiv und schafft damit Redundanz und Übertragungssicherheit. Dazu haben wir verschiedenen Produkte in einem LoRaMesh bereits vernetzt, getestet und kennen die notwendigen Realisierungsattribute.

Ein Prototyp wurde konstruiert, der auf den o.g. Grundlage ein energiesparendes **Meshnetzwerk** aufbauen kann. Weiter wurde in dieser Konstruktion bereits ein Sensor integriert, der die Voraussetzungen für die Erfassung der Bodennahen Gase aufnimmt, auswertet und mittels einer vortrainierten KI entsprechende Waldbrandprognosen liefern kann, - noch bevor es zu einem umfangreichen Waldbrand kommt.

Über Zeitslots, effizient mit einer vorgesehenen Betriebsdauer von bis zu 10 Jahren Messdaten aus dichtbewachsenen Waldgebieten mit einem selbstorganisierenden Meshnetzwerk zu übertragen sind besondere Eigenschaften, die es in dieser Kombination bisher noch nicht gibt. Zur Realisierung sind Kenntnisse im Design, Hard-, Software/Informatik, Funk-/Elektrotechnik essentiell.

## Unterlagen zum Nachbau

Alle aktuellen Unterlagen liegen auf unserem Git-Server unter  
<https://gitea.iotxs.de/RainerWieland/FireNose>  
zum Download bereit.



# Artefakte & Zugänglichkeit

## Quell-Repos:

- Sämtliche Unterlagen sind unter der o.a. URL abgelegt.
- Die Verzeichnisse und deren Inhalte sind zusammengefasst in <https://gitea.iotxs.de/RainerWieland/FireNose/src/branch/main/Info.md> erläutert.
- Gem. den Kernanforderung einer offenen Hardware sind alle editierbare Designdateien veröffentlicht (z. B. Projekt-Dateien, CAD-Modelle) und liegen nicht nur als PDFs/Gerbefiles/etc. ab.
- Für verwendete Komponenten, wie dem BME688-Sensor wird es zukünftig den Ordner mit BSEC2-Integration und der AI-Studio-Projekt/Export-Konfiguration geben. Damit wird das später durchgeführte Training und zur Laufzeit notwendige Modell eindeutig und nachvollziehbar. Die von Bosch bereitgestellte Software ist unter <https://www.bosch-sensortec.com/software-tools/software/bme688-and-bme690-software/> zu finden und kann dort heruntergeladen werden.

## Release/Tag mit reproduzierbarem Stand:

- Unter <https://gitea.iotxs.de/RainerWieland/FireNose/releases> werden die letzten und aktuell gültigen Freigaben veröffentlicht. Derzeit nur zu Testzwecken der Umgebung ist dort ein ProofOfConcept (PoC) hinterlegt (ohne Dokumente und ohne Quellen)
- Versionen der verwendeten Tools, soweit notwendig, werden hier veröffentlicht. Bei der Softwareentwicklung unter VSC werden die verwendeten Bibliotheken mit Version-Nr in der Konfigurationsdatei hinterlegt, um eine fehlerfreie Reproduzierbarkeit zu gewährleisten.

### Beispiel (*Auszug aus platformio.ini*):

```
lib_deps =
  packageName=nanopb/library/Nanopb
  nanopb/Nanopb@0.4.91
  packageName=erriez/library/ErriezCRC32
  erriez/ErriezCRC32@1.0.1
```

## Downloadbare Pakete (Gerbers, STLs, Firmware-Binaries):

- Neben den öffentlich zugängigen Quelldateien werden zusätzlich Dateien bereitgestellt, die es jedem erlauben, die Hardware, übersetzte Firmware, sowie die notwendigen CAD-Dateien (stp für den Gehäusedruck) herunter zu laden, um direkt mit der Produktion beginnen zu können.

## Dokumentations-/Readme-Einstiegspunkt:

- Der zentraler Einstiegspunkt ist direkt unter <https://gitea.iotxs.de/RainerWieland/FireNose> verankert.



Hier stehen alle notwendigen Informationen strukturiert über **readme.md** zur Orientierung bereit.

## OSH-Konformität:

Eine OSHWA-konform ist denkbar, jedoch im aktuellen Stand noch nicht bereit zur Anmeldung.

An offenen Lizenzen werden verwendet:

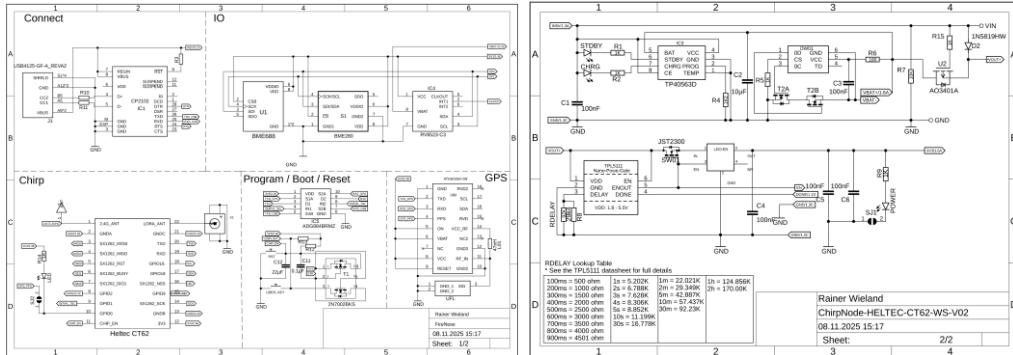
- CERN-OHL-P für Hardware. Diese Variante erlaubt weitgehend freie Nutzung, Modifikation und Distribution der Hardware, solange Urhebervermerke erhalten bleiben.
- Apache-2.0 für Firmware ist vorgesehen. Sie bietet rechtlichen Schutz und Klarheit speziell im Bereich Patente und Änderungen, bleibt dabei aber permissiv. Sie erfordert eine bewusste Beachtung der Anforderungen an Lizenzhinweise und Änderungserklärungen (coming soon). Im Quelltext wird ein entsprechender SPDX-Lizenzheader gesetzt, um die Lizenz des Codes maschinenlesbar und eindeutig erkennbar zu machen.
- CC BY-SA für Dokumentation. Die Creative Commons Lizenz CC BY-SA (Namensnennung - Weitergabe unter gleichen Bedingungen) erlaubt es, ein Werk zu verbreiten, zu verändern, zu remixen und auch kommerziell zu nutzen

# Dokumentation

- Baupläne

Derzeit liegt noch keine endgültige Platine als Prototyp oder für die spätere Nullserie/Produktion vor. Die Baupläne werden während der ersten Inbetriebnahme erstellt. Verbesserungen fließen kontinuierlich in die Dokumentation mit ein.

Die aktuellen Konstruktionsunterlagen wie Stromlaufpläne, und Layoutdateien sind verfügbar und können nach den Lizenzbestimmungen heruntergeladen und genutzt werden.



- Stückliste

Bei der Auswahl der elektronischen Bauteile wurde neben den funktionalen Anforderungen auch auf Verfügbarkeit und Kosten geachtet. Die Stückliste ist derzeit lückenhaft und kann erst nach der Inbetriebnahme finalisiert werden.

- Reproduzierbarkeit

Für den Gehäusedruck wurde ein Material ausgewählt, welches für einen langfristigen Testbetrieb unter Umweltbedingungen geeignet ist (ASA). Daraus ergeben sich Einstellungen für den 3D-Druck (z.B. Temperaturbereich), der u.U. nicht von allen 3D-Druckern eingehalten werden kann. Hierauf wird in der gesonderten Dokumentation zum 3D-Druck hingewiesen.

Für die Produktion und Bestückung der Elektronikplatine wird in einer gesonderten Dokumentation die Umsetzung beschrieben. Häufig ist innerhalb der BOM eine Produzenten-abhängige Anpassung i.F. der Artikel-/Bestellnummer notwendig.

## Weiterentwicklung

Derzeit liegt der Fokus in diesem Projekt auf Funkübertragung und Energieverbrauch. Sofern diese beiden Punkte abschließend untersucht und über den ersten Prototypenlauf bestätigt werden, wird das KI-Modell zur Gasdetektion, die thermische Diffusivität sowie die notwendige Software-Infrastruktur und Migration der Daten von installierten Wärmebildkameras zur Datenverarbeitung weiter entwickelt. Entsprechend sind für die Realisierung der Prototypen finanzielle Mittel notwendig, welche die Inbetriebnahme in einem Testfeld erst ermöglichen.

## Perspektive

Mit der vollständige Einsicht in die Entwürfe und dem Designs wollen wir Vertrauen schaffen und den Nutzern die Möglichkeit geben, die Lösungen nachzuvollziehen zu verstehen, wie und warum es „as is“ funktioniert, ohne auf proprietäre Blackboxes angewiesen zu sein.

Wir wollen damit auch erreichen, dass sich weitere Entwickler für unser Anliegen interessieren und sich aktiv mit ihrer Expertise an aktuellen Energie-, Umwelt-, und Klimaprojekten beteiligen. Damit wird es möglich, unsere Hardware- und Software, unabhängig von großen Herstellern oder teuren Lizenzen, zu entwickeln, zu produzieren und zu modifizieren. Dies senkt die Entwicklungskosten und macht unsere eingesetzte Technologie zugänglicher, auch für kleinere Organisationen, Hobbyisten und andere Projekte.

Weiter ist denkbar, dass unser offenes Hard- und Softwarewareprojekt als Lernmaterialien genutzt werden kann, es verbessert unserer Meinung nach das technische Verständnis und ermöglichen praxisnahe Ausbildung in Schule, Betrieb, Forschung und Lehre.

## Nachhaltigkeit & Ressourceneffizienz

Unsere Lösung ist so ausgerichtet, dass sie, dort wo es möglich ist (zum Beispiel in Waldlichtungen), mit erneuerbarer Energie versorgt wird, oder mit einer Batterie über einen langen Zeitraum (bis zu 10 Jahren) wartungsfrei genutzt werden kann. Bei unserer Lösung verzichten wir auf aufwändige Montage und Spezialwerkzeug. Für PV- und Batterieeinsatz wird identische Hardware eingesetzt, was die Variantenvielfalt reduziert, späterer Masseneinsatz vereinfacht und so die Rüst-Kosten für eine größere Produktion bei Sammelbestellungen, etc. ebenfalls senkt.

Das Gehäuse ist in Form und Größe so konstruiert, das es selbst auf einem kleinen 3D-Drucker mit einer Heizbettgröße von nur 18x18cm (z.Bsp. Bambu mini A1) komplett auf einmal gedruckt werden kann. Die Komponenten werden bei der Montage gesteckt und durch gedruckte Gewinde miteinander verschraubt. Es sind keine Fremdprodukte wie Schrauben, Klemmen etc. notwendig. Das Gehäuse entstammt ursprünglich aus einem anderen Projekt (Urban-Climate) und wurde geringfügig adaptiert.



## Technische Umsetzung – Ergebnis

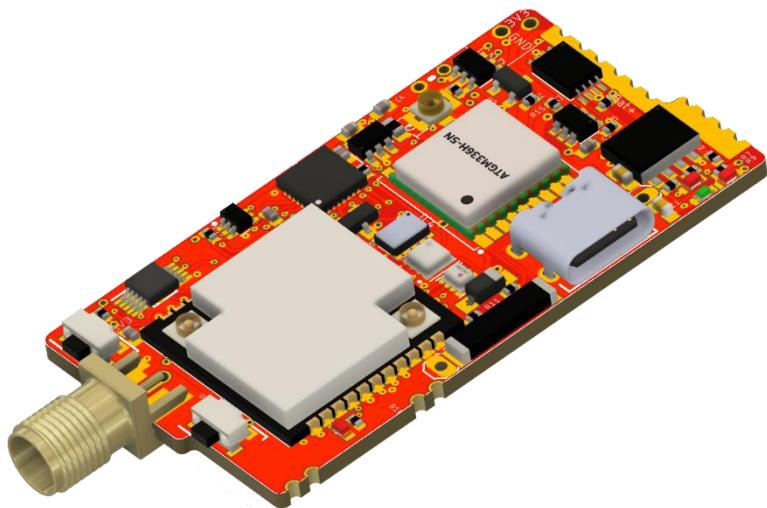
Wie bereits oben erwähnt, liegt zunächst der Fokus auf der **Funkübertragung und der Energieeffizienz**.

Bei der Realisierung des Funk-Meshnetzwerkes wurde zusätzlich ein weiterer Prototyp entworfen, mit dem die Funktechnologie genauer untersucht werden kann. Insbesondere soll eine Lösung erarbeitet werden, mit der die Zeitslots, in der das Meshnetz synchron aktiv ist, optimiert wird.

Der **weitere Prototyp** beheimatet zusätzlich eine Echtzeituhr (RTC) mit Alarmfunktion zum Aktivieren der weiteren Hardware (WakeUp). Die ausgewählte RTC hat eine typische Stromaufnahme von 130nA, und gilt als eine der energiesparendsten RTCs überhaupt (Typ: RV-8523-C3). Sie soll später in den Sensoren mit integriert werden.

Weiter wurde ein GPS-Decoder implementiert, mit dem die aktuelle Uhrzeit auf wenige Nanosekunden genau erfasst werden kann (PPT). Hiermit wird es möglich, die Zeitslots über einen Supervisor (Hub/Relaisstation) in notwendigen Abständen zu synchronisieren.

Zur Entwicklung wurde die Platine über einen FTDI-Controller (USB  $\leftrightarrow$  Serial) erweitert. Damit wird die Programmierung und das Monitoring während der Entwicklungsphasen wesentlich vereinfacht. Über eine USB-C Schnittstelle kann zusätzlich die Platine parametrisiert und modifizierte KI-Modelle für den Umwelt-Chip hineingeladen werden.



## Technische Umsetzung – Methodik

Zunächst wurde auf vorhandene und verfügbare LoRaMesh-Platinen zurückgegriffen, um die Funktionsweise zu durchdringen und den Funkbetrieb in Gänze zu verstehen. Ein Teil davon konnte mit eigener, modifizierter Firmware betankt werden. Ein anderer Teil wurde aufgrund von Programmier-Schwierigkeiten und der komplexen Handhabbarkeit aussortiert.

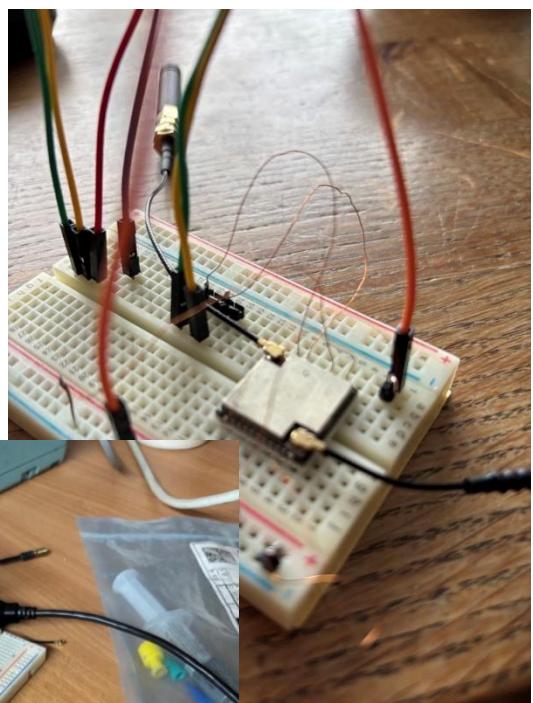
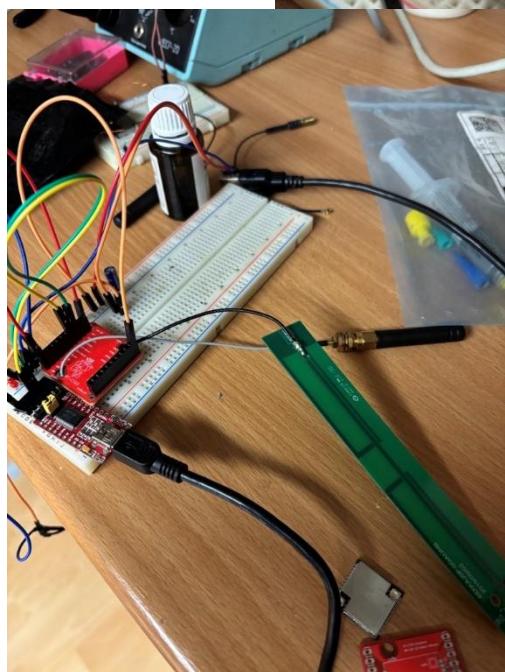
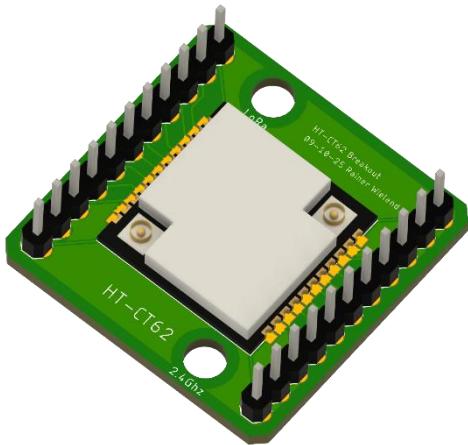
Bei der internationalen UKW-Tagung im September wurde eigens für das Projekt ein **Workshop** und ein **Vortrag** gehalten.

Die Teilnehmer (eine IT-Klasse aus der dualen Berufsausbildung) des Workshops hatten bisher nur theoretische Grundlagen zu Meshnetzwerken und konnten mit den Platinen erste praktische Erfahrungen sammeln. Weiter gab uns dieser Workshop eine Gelegenheit, unser Fortbildungssetting zu testen und für eine spätere Dokumentation zu reflektieren.

Der Vortrag zielte auf Teilnehmer (Funkamateure) die sich im Funkumfeld bereits sehr gut auskennen und Nutzer von LoRaMesh bzw. hessenMESH. Ziel war es, mit Ihnen in Dialog zu treten, um weitere Erkenntnisse (Vor- und Nachteile der bekannten Lösungen) zu gewinnen und sich mit der Idee einem fachkundigen Publikum zu stellen.

Die Ergebnisse flossen in die Entwicklung der eigenen Platine mit ein. Für den zweiten Prototypen (wie oben angegeben), wurde gezielt ein Controller ausgewählt, für den es bereits ein Binary für Meshtastic gibt. Dieser wurde „baremetal“ auf einem Breadboard über Bonddrähten für einen ersten Test in Betrieb genommen.

Im Anschluss wurde ein Breakout entworfen, damit weitere Soft- und Hardwaretest realisiert werden konnten.



## Nächste Schritte

Die Platinen für den Prototypen und das Devboard sind in kleinen Stückzahlen bestellt und auf dem Weg zur Produktion. Wir werden diese wie folgt in Betrieb nehmen:

- Sichtkontrolle der unbestückten Prototyp-Platine
  - Lötanschlüsse für PV-Modul,
  - Beschriftung Oberseite
  - Beschriftung Unterseite
- Elektronischer Test der Prototyp-Platine
  - Lötbrücke für Akku-Ladeelektronik setzen
  - Anschluss an eine Festspannungsquelle (5V mit Strombegrenzung) und Messung der Stromaufnahme
  - Anschluss eines Lion-Akkus und Messung der Stromaufnahme bis zur oberen Abschaltung (Grenzspannung des Akkus)
  - Kontrollierte Entladung des Lion-Akkus und Messung der Spannung bis zum unteren Grenzwert (Abschaltung zum Akkuschutz muss einsetzen)
  - Lötbrücke für Akku-Ladeelektronik entfernen
  - Lötbrücke für Batterie (3,7V 19Ah) setzen
  - Anschluss an eine Festspannungsquelle (3,7V mit Strombegrenzung) und Messung der Stromaufnahme
  - Testsoftware auf µController laden und Stromaufnahme im Betrieb und UltraDeep-Mode überprüfen
- Sichtkontrolle der unbestückten Devboard-Platine
  - Lötanschlüsse für PV-Modul,
  - Beschriftung Oberseite
  - Beschriftung Unterseite
- Elektronischer Test der Devboard-Platine
  - Lötbrücke für Akku-Ladeelektronik setzen
  - Anschluss an eine Festspannungsquelle (5V mit Strombegrenzung) und Messung der Stromaufnahme
  - Anschluss eines Lion-Akkus und Messung der Stromaufnahme bis zur oberen Abschaltung (Grenzspannung des Akkus)
  - Kontrollierte Entladung des Lion-Akkus und Messung der Spannung bis zum unteren Grenzwert (Abschaltung zum Akkuschutz muss einsetzen)
  - Testsoftware auf µController laden und Stromaufnahme im Betrieb und UltraDeep-Mode überprüfen
- Allgemeine Entwicklung/Inbetriebnahme
  - Software-Bibliothek zur Kommunikation mit BME688 erstellen
  - Software-Bibliothek zur Kommunikation mit dem GPS-Modul erstellen
  - Software-Bibliothek zur Kommunikation mit dem RTC-Modul erstellen
  - DeepSleep über Timertrigger realisieren
  - Synchronisationssoftware entwickeln
  - LoRaMesh-Software auf Basis von Meshtestic implementieren und testen
  - LoRaMesh-Software um TimeSync erweitern