

# Alarm-Board Anleitung

Das Alarm-Board wurde für Berufsfeuerwehrtage bei Jugendfeuerwehren entwickelt. Es simuliert Alarmierungen, bei denen Einsätze und Fahrzeuge über eine zentrale Weboberfläche verwaltet werden können.

Über die

Weboberfläche können Alarne manuell ausgelöst und zugehörige Einsatzfahrzeuge zugewiesen werden.

Die Anwendung bietet auch die Möglichkeit, über Funk (Meshtastic) Alarmierungen zu versenden.

In dieser Anleitung wird die Installation und Nutzung des Alarm-Boards erklärt.

## Funkbasierte Alarmierung mit Meshtastic

Mit dem Meshtastic-Gerät können Alarne nicht nur über das Netzwerk, sondern auch über Funk versendet werden. Dies erweitert die Flexibilität des Systems.

P1

Alarm ausgelöst! Einsatz:  
Testalarm ++ Testalarm ++,  
Ort: Zehdenick Falterstrasse  
15, Fahrzeuge: MTF

10:15

P1

Nachalarm! Einsatz:  
Testalarm ++ Testalarm ++,  
Ort: Zehdenick Falterstrasse  
15, Nachalarmierte  
Fahrzeuge: GW

10:16

## Nutzung des Alarm-Boards

Die Anwendung wird über die Weboberfläche gesteuert, auf die man über die IP-Adresse des Raspberry Pi zugreifen kann. Hier können Einsätze manuell eingegeben und die zugehörigen Fahrzeuge ausgewählt werden.



# Alarm Monitor



The image shows a red fire truck with 'FEUERWEHR' and '012' markings parked in front of a building with a sign that says 'FEUERWEHR'. The truck is a MAN model. A mobile device screen is overlaid on the image, displaying an 'Einsatz-Eingabe' (Emergency Input) form. The form includes fields for 'Einsatz:' (Brand klein), 'Ort (Adresse):' (Zehdenick Poyenweg), and coordinates. A keyboard is visible at the bottom of the screen.

Einsatz-Eingabe  
Samstag, 5.10.2024, 15:01:07

Einsatz:  
Brand klein

Ort (Adresse):  
Zehdenick Poyenweg

Koordinaten (Lat): Koordinaten (Lon):

Poyenweg Porenwege Porenwegen ↗  
q w e r t z u i o p  
a s d f g h j k l  
y x c v b n m ↵  
!@# . , ? !? ↷

# Alarm Monitor

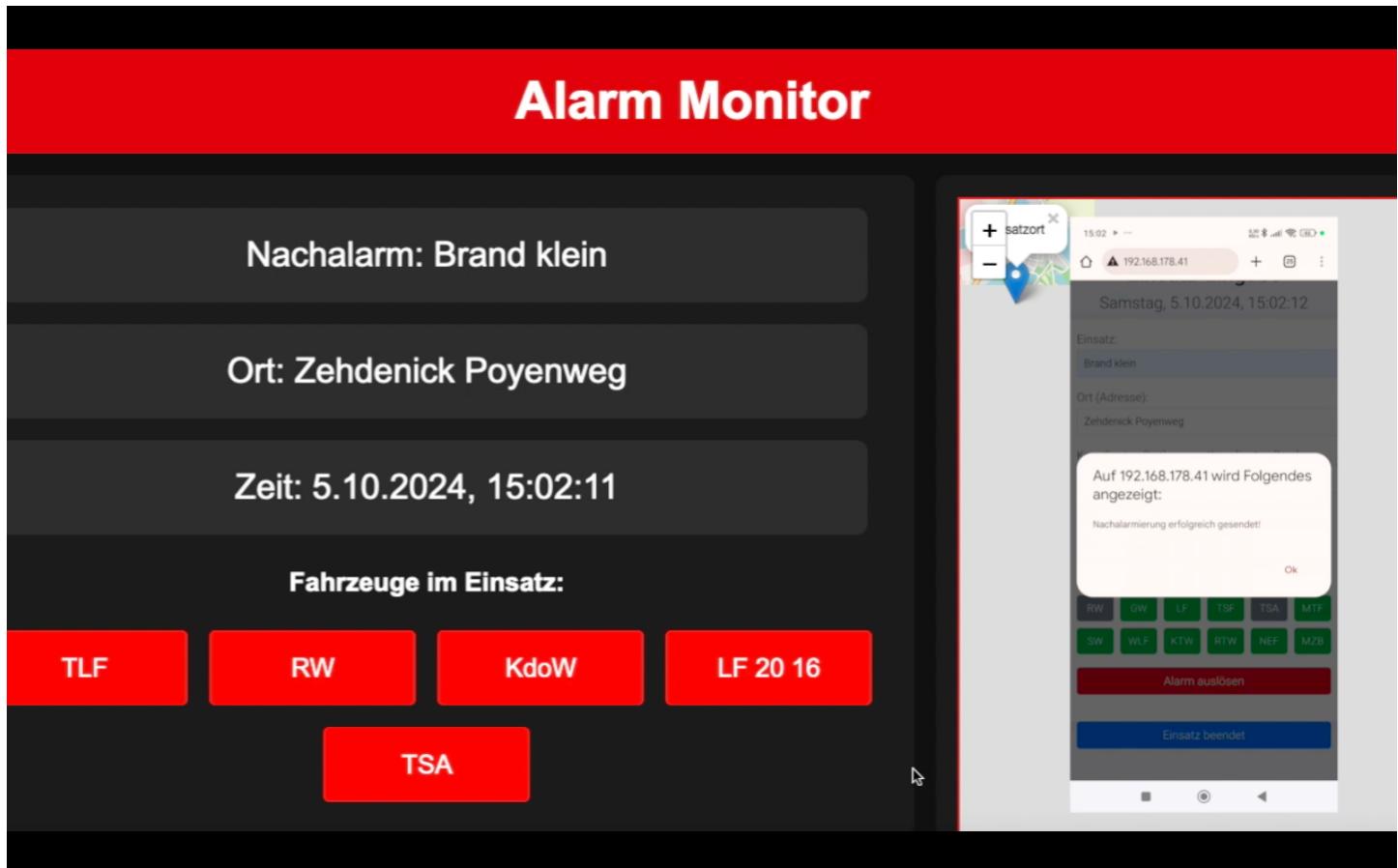
The image shows a red MAN fire truck parked on a grassy field. The truck has 'FEUERWEHR' and 'MAN' written on the front. A mobile phone screen is overlaid on the right side of the image, displaying the 'Einsatz-Eingabe' app. The app interface includes:

- Timestamp: 15:01 Sam ...
- Date: Samstag, 5.10.2024, 15:01:22
- Incident Type: Einsatz: Brand klein
- Location: Ort (Adresse): Zehdenick Poyenweg
- Coordinates: Koordinaten (Lat): 52.986435 | Koordinaten (Lon): 13.346919
- Vehicle Selection Grid:

ELW	KdoW	TLF	TLF 9000	DLK	LF 20/16
RW	GW	LF	TSF	TSA	MTF
SW	WLF	KTW	RTW	NEF	MZB
- Action Buttons: 'Alarm auslösen' (red) and 'Einsatz beendet' (blue)
- Photo Credit: Foto: L
- Watermark: © Lars Heuer

## Alarmbestätigung und Nachalarm

Nach dem Auslösen eines Alarms wird der Einsatz auf dem Alarm-Monitor angezeigt. Es besteht die Möglichkeit, einen Nachalarm zu versenden und zusätzliche Fahrzeuge hinzuzufügen.



## Alarmmeldungen auf einem mobilen Gerät

Auf mobilen Geräten wird die Alarmmeldung wie folgt angezeigt. Dies umfasst die Einsatzbeschreibung, den Ort und die alarmierten Fahrzeuge.



# Schritt-für-Schritt Anleitung zur Installation

## Schritt 1: Installationsskript herunterladen und ausführen

1. Öffne ein Terminal auf deinem Raspberry Pi.
2. Lade das Installationsskript herunter oder kopiere den folgenden Inhalt in eine Datei namens `install_script.sh`:

```
#!/bin/bash
```

```
# Update and upgrade system packages
```

```
sudo apt update && sudo apt upgrade -y
```

```
# Install Python and pip
```

```
sudo apt install -y python3 python3-pip
```

```
# Install necessary Python packages
```

```
pip3 install flask flask-socketio gtts meshtastic
```

```
# Install node.js and npm (for socket.io and axios)
```

```
curl -fsSL https://deb.nodesource.com/setup_14.x | sudo -E bash -
```

```
sudo apt install -y nodejs
```

```
# Install necessary JavaScript packages
```

```
sudo npm install axios socket.io
```

```
# Install Leaflet for map display
```

```
sudo npm install leaflet
```

```
# Create directories for logs and static files
```

```
mkdir -p logs
```

```
mkdir -p static/bilder
```

```
# Download Leaflet CSS and JS files
```

```
sudo wget https://unpkg.com/leaflet@1.7.1/dist/leaflet.js -P static
```

```
sudo wget https://unpkg.com/leaflet@1.7.1/dist/leaflet.css -P static
```

```
# Set permissions for web server access to static files
```

```
sudo chmod -R 755 static
```

### 3. Mache das Skript ausführbar:

```
chmod +x install_script.sh
```

### 4. Führe das Skript aus:

```
./install_script.sh
```

## Schritt 2: Anwendung starten und auf die Weboberflächen zugreifen

Nach der erfolgreichen Installation kannst du die Anwendung im Terminal mit dem folgenden Befehl starten:

```
python3 app.py
```

**Die Alarm-Eingabe erfolgt auf:**

**http://<Deine-Raspberry-Pi-IP>:5000**

**Die Alarm-Monitor-Seite kann aufgerufen werden unter:**

**http://<Deine-Raspberry-Pi-IP>:5000/monitor.html**