

# Digitale Sirenensteuerung - Phönix

## Benutzermanual



### Revision History

Rev.	Anmerkungen	Autor	Seiten	Datum
01	Dokument erstellt	DI Auer Andreas	alle	23.06.15

## Inhalt

1 Beschreibung.....	3
2 Komponenten.....	3
2.1 Backplane.....	3
2.2 CPU Hauptplatine.....	5
3 Bedien- und Anzeigeelemente.....	5
3.1 Front.....	5
3.1.1 Taster.....	5
3.1.2 LEDs.....	6
3.1.3 USB und Netzwerk.....	7
3.2 Backplane.....	7
3.2.1 Ausgänge (8).....	8
3.2.2 Eingänge (9).....	8
3.2.3 Spannungsversorgung.....	9
3.2.4 16-poliger Wannenstecker (7).....	9
3.2.5 9-poliger SUB-D (6).....	9
4 Konfiguration.....	9
4.1 Allgemeine Einstellungen.....	10
4.2 Eingänge.....	11
4.3 Ausgänge.....	12
4.4 Fixtexte.....	13
4.5 Aktionen.....	13
4.5.1 Aktion Funk.....	14
Ausgänge.....	14
Verzögerung.....	15
Filter.....	15
Pager.....	16
Rückmeldung.....	17
4.5.2 Aktion Eingang.....	18
4.5.3 Aktion Taste.....	19
4.5.4 Aktion Spannung.....	19
4.5.5 Aktion Watchdog.....	20
4.5.6 Aktion PC.....	21
4.6 Sirenenprogramme.....	22
4.7 Einstellungen.....	23
4.8 Steuerung.....	24
4.8.1 Logging.....	24
4.8.2 Status.....	25
4.8.3 Befehle.....	26
5 Anhang.....	27
5.1 Schaltplan und Lageplan CPU Core Board.....	27
5.2 Schaltplan und Lageplan Backplane.....	27

## 1 Beschreibung

Die digitale Funksirenensteuerung „Phönix“ ist für den Einsatz als Sklavenempfänger entworfen worden. Entsprechend sind auch nur 4 Tasten zur Bedienung vorgesehen.

Das angeschlossene Tait TM8100 Funkgerät stellt die Verbindung zur Luftschnittstelle her und demoduliert die empfangenen Funksignale. Die Phönix Steuerung kann dann das vom Funkgerät demodulierte Basisbandsignal weiterverarbeiten und auswerten.

Für die Übertragung der digitalen Telegramme wird die 4-Level FSK (4LFSK) Modulation verwendet. Das 4LFSK modulierte Signal wird mit dem FX919B von CML Microcircuits ausgewertet. Der Inhalt des Alarmierungstelegramms wird anschließend von der CPU verarbeitet und entsprechende Aktionen ausgeführt.

Neben der Ansteuerung der Sirene über einen dedizierten Ausgang, bietet die Steuerung noch weitere 7 potentialfreie Relaisausgänge sowie 8 potentialfreie Eingänge. Die einzelnen Ausgänge können beliebigen Events zugeordnet werden und sowohl zeitlich verzögert geschaltet werden als auch unterschiedliche Schaltsequenzen (Impuls, Toggle, Level) durchführen. Die Eingänge können unterschiedliche Aktionen auslösen.

Der Sklavenempfänger ist auch in der Lage, digitale POCSAG Pager zu alarmieren. Dazu werden die Adressen der Pager sowie fixe Texte hinterlegt. Bei einer Funkalarmierung kann auch der mitgesendete Text an den Pager weitergeleitet werden.

Für die Zukunft ist die Steuerung auch für den Anschluss eines TETRA Funkgerätes über die PEI Schnittstelle vorbereitet.

## 2 Komponenten

In Abbildung 1 ist der Aufbau der Steuerung anhand eines Blockdiagramms dargestellt. Der mittlere große Block stellt den 19 Zoll Baugruppenträger dar mit der Backplane an der Hinterseite und der Hauptprozessorplatine.

### 2.1 Backplane

Die Backplane ist für die Spannungsversorgung der einzelnen Komponenten verantwortlich. Im normalen Betrieb liefert ein 230VAC Netzteil die 12V Betriebsspannung an die Backplane. Ein optional angeschlossener Bleigel Akku sorgt für den Fall eines Stromausfalls für den weiteren störungsfreien Betrieb der Anlage. Wenn die Netzspannung verfügbar ist, wird der angeschlossene Akku auch geladen. Den korrekten Ladevorgang übernimmt ein speziell dafür vorgesehener IC von Texas Instruments (UC3906).

Zusätzlich gibt es noch einen weiteren 12V Eingang, der in Abbildung 1 mit „+12V AUX“ gekennzeichnet ist. Über diesen kann die Anlage ebenfalls versorgt werden. Dieser Eingang wird aber nicht überwacht und auch eine Akkuladung (falls ein zusätzlicher Akku angeschlossen wird), erfolgt nicht.

Die Überwachung der Spannungseingänge „+12V Netz“ und „+12V Pb-Akku“ übernimmt ein zusätzlicher Baustein. Dieser erkennt auch, ob die angelegte Spannung sich in einem gültigen Bereich befindet. Für den angeschlossenen Akku ist damit eine Unterspannungserkennung realisiert, um den Akku vor Tiefentladung zu schützen.

Auf der Backplane befinden sich auch die potentialfreien Relais Ausgänge und die potentialfreien Eingänge. Es sind 8 Ausgänge vorgesehen, wobei Ausgang 1 speziell für die Ansteuerung des Sirenenrelais vorgesehen ist. Parallel zu jedem Relaisausgang ist eine LED angeordnet, die den Zustand des Ausganges anzeigt (leuchtet die LED ist das Relais angezogen).

Für den Anschluss von externen Signalgebern (z.B. Brandmelder) sind 8 potentialfreie Eingänge vorhanden.

Auf der Backplane befindet sich auch der 16-polige Wannenstecker für das Tait TM8100 Funkgerät. Die Signale des Funkgerätes werden direkt an die CPU Platine durchgeschliffen. Optional kann das Funkgerät auch direkt mit der CPU Platine verbunden werden.

Ein 9-poliger SUB-D Stecker ist für den Anschluss eines TETRA Funkgerätes verfügbar. Die PEI Schnittstelle wurde softwareseitig aber noch nicht implementiert.

Die Funkgeräte werden ebenfalls von der Backplane versorgt. Es sind zwei „12V“ Ausgänge vorhanden. Einer zur Versorgung des analogen Funkgerätes, der andere für das TETRA Funkgerät. Für die Stromversorgung des analogen Tait Funkgerätes wurde auch eine Strommessung implementiert. Damit kann gemessen werden, ob das Funkgerät auch genügend Strom beim Senden aufnimmt.

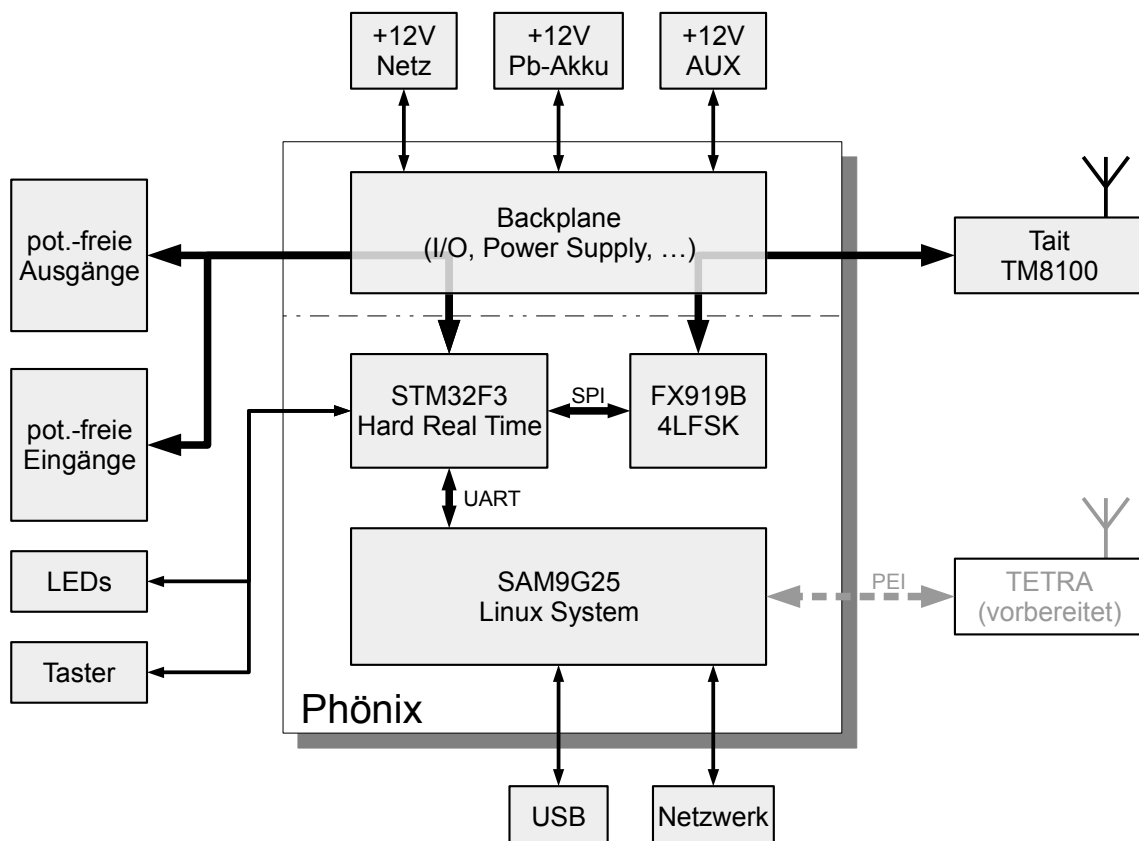


Abbildung 1: Blockdiagramm

## 2.2 CPU Hauptplatine

Die CPU Hauptplatine wird im 19 Zoll Baugruppenträger von vorne eingeschoben und steckt über einen 96-poligen Stecker auf der Backplane. Auf dieser Platine befinden sich die gesamten Verarbeitungseinheiten. Der Auswertechip FX919B für die analogen 4LFSK Signale ist an den Mikrocontroller STM32F407 angeschlossen. Dieser Mikrocontroller ist vor allem für die harten Echtzeitaufgaben, wie die Auswertung des Alarmierungstelegramms und das Erzeugen von POCSAG Nachrichten, verantwortlich. Außerdem werden die Ausgänge, die Eingänge, die LEDs und die Taster an der Frontplatte über den Controller verwaltet.

Die Verarbeitung aller Events erfolgt dann unter Linux, welches auf dem SAM9G45 Board läuft. Dieses Board lädt das Linux Betriebssystem von einer Micro SD Karte, die sich direkt auf der CPU Platine befindet. Über eine serielle Schnittstelle ist das Linux Board mit dem Echtzeit Mikrocontroller verbunden. Wenn ein Alarmtelegramm ausgewertet, ein Eingang aktiv oder ein Taster an der Frontplatte gedrückt wurde, wird dies dem Linux System mitgeteilt und es werden entsprechende Aktionen ausgeführt.

Die Netzwerkschnittstelle und die beiden USB Ports an der Frontseite der Sirenensteuerung werden direkt vom Linux Board verwaltet. Über die Netzwerkschnittstelle wird das System auch konfiguriert.

## 3 Bedien- und Anzeigeelemente

An der Frontplatte befinden sich alle wichtigen Bedien- und Anzeigeelemente. An der Rückseite des 19 Zoll Baugruppenträgers ist die Backplane montiert, auf der sich die Eingänge und Ausgänge sowie die verschiedenen Anschlüsse zur Stromversorgung befinden.

### 3.1 Front

An der Frontseite der Sirenensteuerung sind 4 Taster für die Bedienung, 8 LEDs zur Visualisierung des aktuellen Zustandes sowie die USB und Netzwerkschnittstelle angeordnet (siehe Abbildung 2).

#### 3.1.1 Taster

Es sind 4 Taster zur Bedienung vorhanden:

- **FEUER** (mit Schutzkappe)  
Dieser Taster löst vor Ort das Sirenenprogramm „Feuer“ aus, wenn dies mit der PC Software entsprechend konfiguriert wurde.
- **SIRENE HAND**  
Dieser Taster dient der „Handsteuerung“ der Sirene. Solange der Taster gedrückt wird, ist auch das Sirenenrelais aktiv. Dieser Taster wirkt sich direkt auf den Sirenenausgang aus und wird nicht durch den Prozessor verarbeitet. Das bedeutet, dass der Sirenenausgang auch im Falle eines defekten Prozessors gesteuert werden kann.
- **SIRENE AUS**  
Ähnlich wie der „SIRENE HAND“ Taster wirkt sich auch dieser direkt auf den Sirenenausgang aus. Solange dieser Taster gehalten wird, ist der Sirenenausgang inaktiv.

- **TEST**

Dieser Taster kann verwendet werden, um ein Testsignal zu aktivieren. Dies muss ebenfalls mit der PC Software entsprechend konfiguriert werden.

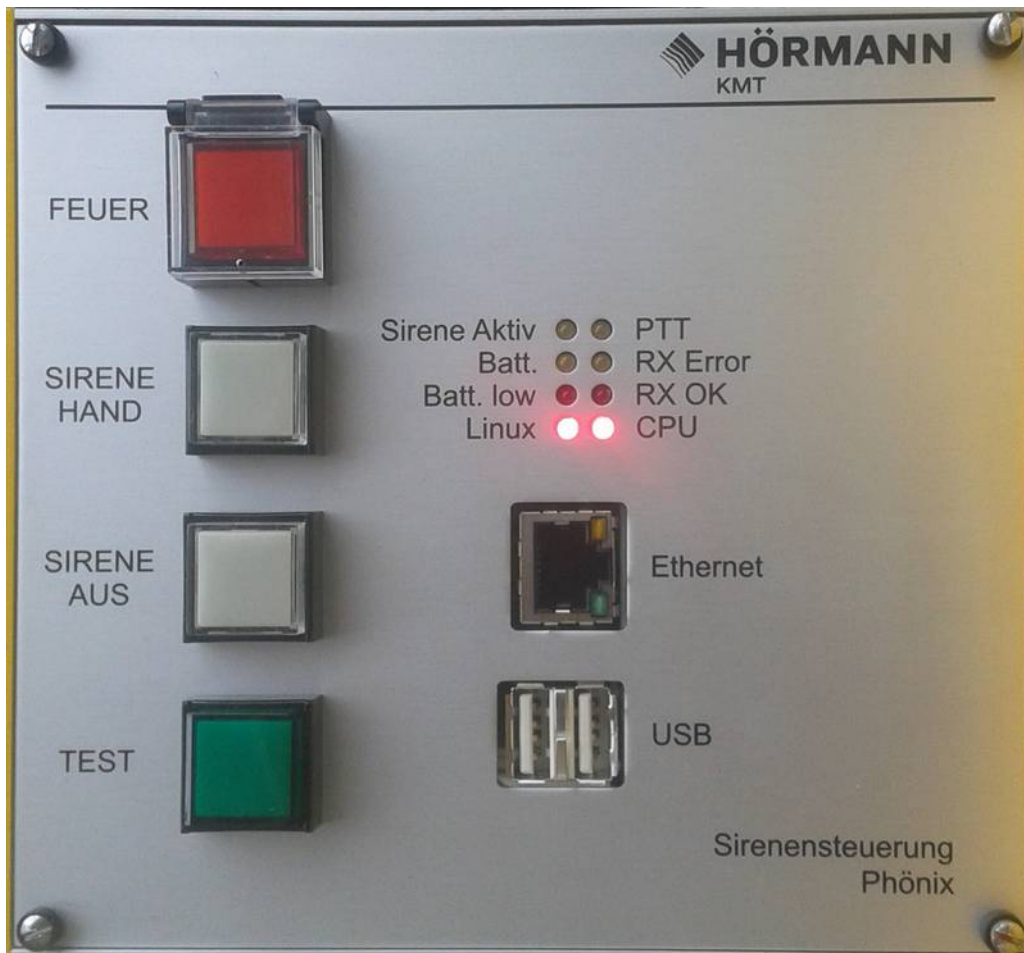


Abbildung 2: Frontplatte mit Bedienelementen

### 3.1.2 LEDs

Die 8 LEDs dienen zur Visualisierung des aktuellen Anlagenzustands.

- **CPU**  
Die CPU LED blinkt, wenn der Mikrocontroller (STM32F3) gestartet wurde und läuft.
- **Linux**  
Die „Linux“ LED leuchtet, wenn das Linux System gebootet hat und wenn die Kommunikation zwischen dem Mikrocontroller und dem Linux System funktioniert.
- **RX OK**  
Diese LED leuchtet, wenn ein 4LFSK Telegramm korrekt und ohne Fehler empfangen wurde. Der Zustand bezieht sich immer auf das zuletzt empfangene Telegramm.

- **RX Error**  
Diese LED leuchtet, wenn beim Empfang des 4LFSK Telegramms ein Fehler aufgetreten ist (z.B. CRC Error)
- **PTT**  
Die PTT LED zeigt an, wenn die PTT Leitung des Funkgerätes aktiv wird und das Funkgerät zu senden beginnt. Die PTT LED wird vom Mikrocontroller gesteuert.
- **Batt.**  
Die „Batt.“ LED leuchtet, wenn keine Netzspannung vorhanden ist.
- **Batt. low**  
„Batt. low“ zeigt an, dass der Akkuladestatus niedrig ist und die Anlage bald abgeschaltet wird, um den Akku vor Tiefentladung zu schützen.
- **Sirene Aktiv**  
Diese LED zeigt den Status des Sirenenausgangs an. Die LED wird gleichzeitig mit dem Sirenenausgang aktiv bzw. inaktiv. Die „Handsteuerung“ der Sirene wirkt sich nicht auf diese LED aus.

### 3.1.3 USB und Netzwerk

Die beiden USB Schnittstellen sind derzeit noch ohne weitere Funktion und dienen als mögliche Erweiterungsschnittstellen für die Zukunft.

Die Netzwerkschnittstelle (Ethernet) wird verwendet, um die Anlage zu konfigurieren. Es handelt sich hierbei um eine standardisierte Schnittstelle mit dem üblichen TCP/IP Protokoll. Die Schnittstelle ist somit kompatibel mit den üblichen Netzwerkgeräten.

Zur Herstellung der Verbindung mit dem PC/Laptop reicht ein einfaches Netzkabel (ein Crossover Kabel ist nicht notwendig).

## 3.2 Backplane

Auf der Backplane (Abbildung 3) befinden sich alle Klemmen für die Ein- und Ausgänge sowie die Anschlüsse für die Spannungsversorgung und die Funkgeräte.

Folgende Anschlüsse sind in Abbildung 3 gekennzeichnet:

1. +12V vom 230VAC Netzteil
2. Bleigel Akku Anschluss (Batterie wird geladen)
3. Zusätzlicher +12V Versorgungsanschluss ohne Überwachung (BAT-AUX)
4. +12V für Funkgerät mit Strommessung (max. 4A)
5. +12V für optionales TETRA Funkgerät
6. 9-poliger SUB-D Stecker mit PEI Schnittstelle
7. 16-poliger Wannenstecker für Tait TM8100 Anschluss
8. Sirenenausgang und 7 weitere potentialfreie Ausgänge
9. 8 potentialfreie Eingänge

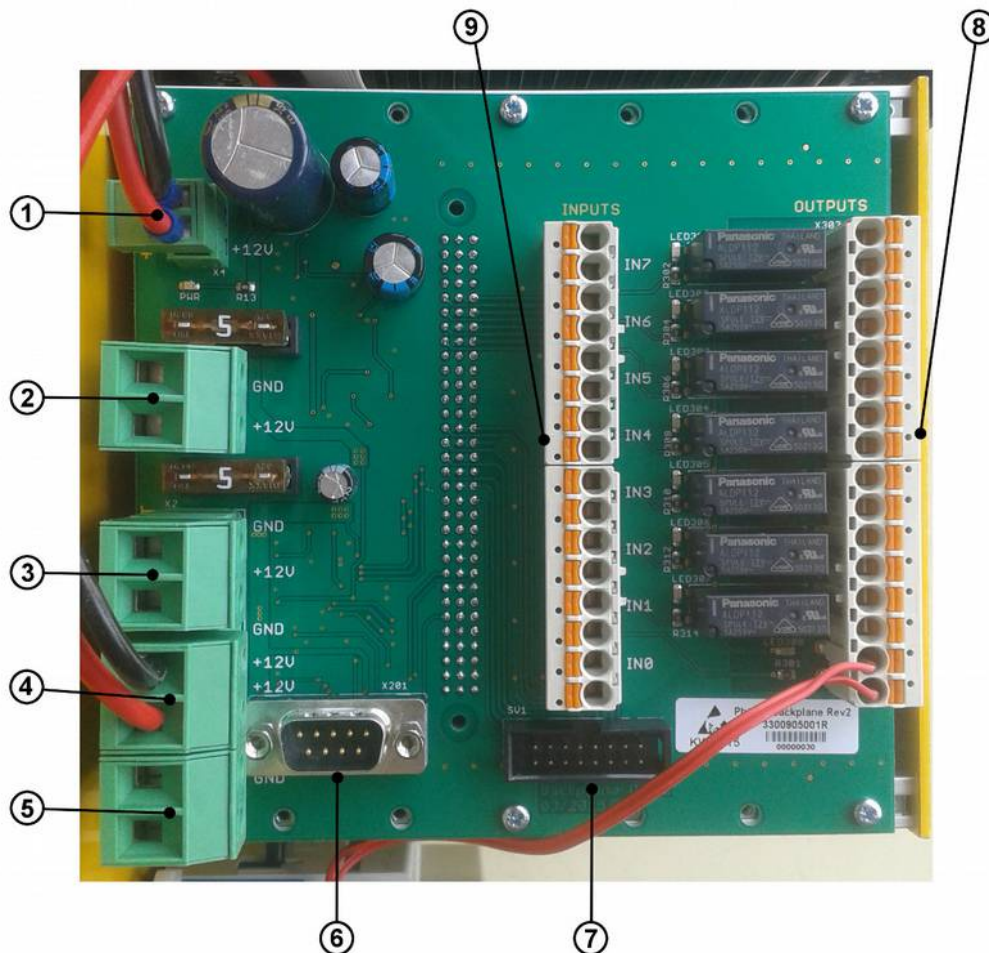


Abbildung 3: Backplane

### 3.2.1 Ausgänge (8)

Es sind 8 Ausgänge vorhanden, wobei der Ausgang 0 (bezeichnet mit „SIR“) ein externes Sirenenrelais steuert. Die weiteren Ausgänge sind mit OUT1 bis OUT7 bezeichnet und als Relaisausgang ausgeführt. Die verwendeten Relais können maximal 5A bei 30VDC oder 277VAC schalten.

Parallel zu jedem Relaisausgang ist auf der Backplane auch eine LED zur Visualisierung des Zustandes angeordnet.

### 3.2.2 Eingänge (9)

Die 8 potentialfreien Eingänge bezeichnet mit IN0 bis IN7 dienen zum Anschluss von externen Signalgebern z.B. eines Brandmelders. Die Eingänge werden softwaremäßig mit einem einstellbaren digitalen IIR Filter vorverarbeitet. Über die PC Software kann die Empfindlichkeit der Eingänge definiert werden.



### 3.2.3 Spannungsversorgung

Auf der Backplane sind mehrere Anschlüsse für die Versorgung der Steuerung und der Funkgeräte vorhanden.

- **+12V (1)**  
Der Eingang „+12V“ dient als Anschluss für das 12V Netzteil, das im Baugruppenträger verbaut ist.
- **BAT (2)**  
Der „BAT“ Anschluss wird verwendet, um einen optionalen 12V Bleigel Akku anzuschließen, der bei vorhandener Netzspannung auch geladen werden soll. Der Akku dient als Notstromversorgung beim Ausfall der Netzspannung.
- **AUX-BAT (3)**  
„AUX-BAT“ ist ein weiterer Anschluss für eine Batterie. Dieser Eingang wird aber nicht geladen und auch nicht weiter überwacht.
- **RADIO (4)**  
Die Klemme mit der Bezeichnung „RADIO“ ist für die Versorgung des Funkgerätes gedacht (12V). Die Stromaufnahme des Funkgerätes wird hier gemessen und dient als Referenz, ob das Funkgerät beim Senden auch genügend Strom aufgenommen hat.
- **TETRA (5)**  
Die „TETRA“ Klemme wurde für die Stromversorgung eines Tetra Funkgerätes vorgesehen. Eine Strommessung ist nicht vorgesehen.

### 3.2.4 16-poliger Wannenstecker (7)

Der 16-polige Wannenstecker wird mit dem 15-poligen SUB-D Stecker am analogen Tait TM8100 Funkgerät verbunden.

### 3.2.5 9-poliger SUB-D (6)

Auf dem 9-poligen SUB-D Stecker befindet sich die optionale PEI Schnittstelle zum Anschluss eines TETRA Funkgerätes.

## 4 Konfiguration

Die Konfiguration für die Sirenensteuerung wird mit der PC Software „Phönix Konfiguration“ erstellt und an die Anlage übertragen. Die Software ist mit Microsoft Windows XP, Microsoft Windows 7, Microsoft Windows 8 sowie mit Linux kompatibel.

Nachdem die Software gestartet wurde, werden die allgemeinen Einstellungen angezeigt, die konfiguriert werden können (Abbildung 4).

Wie auch in anderen PC Programmen üblich, können die Einstellungen über das Menü „Datei“ gespeichert und geladen werden. Ebenfalls kann eine neue Konfiguration mit den hinterlegten Standardwerten erzeugt werden.

## 4.1 Allgemeine Einstellungen

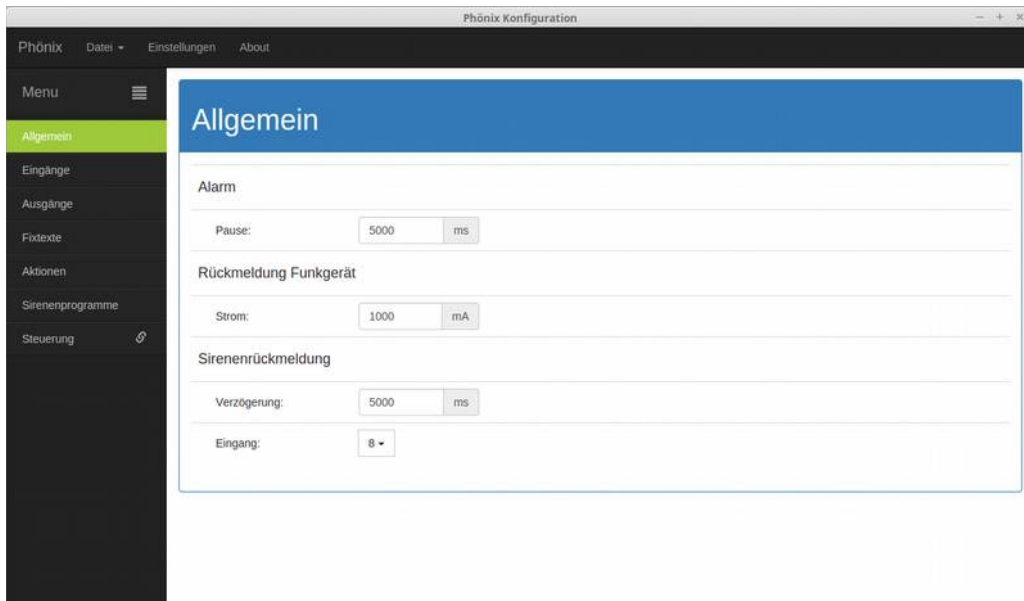


Abbildung 4: Allgemeine Einstellungen

In den allgemeinen Einstellungen finden sich folgende Einstellungen:

- **Alarm**
  - **Pause:** Spezifiziert die Zeit die zwischen zwei Alarmierungen gewartet wird (Default: 5000ms).
- **Rückmeldung Funkgerät**
  - **Strom:** Legt den Schwellwert für den Strom fest, den das Funkgerät beim Senden aufnehmen soll. Wird der Schwellwert überschritten, entspricht dies einer erfolgreichen Rückmeldung des Funkgerätes.
- **Sirenenrückmeldung**
  - **Verzögerung:** Legt den Zeitpunkt für die Auswertung des Sirenenrückmeldeeingangs fest. Die zeitliche Verzögerung wird vom Start der Sirene weg gemessen.
  - **Eingang:** Legt den Eingang fest, der für die Sirenenrückmeldung eingelesen wird.

## 4.2 Eingänge

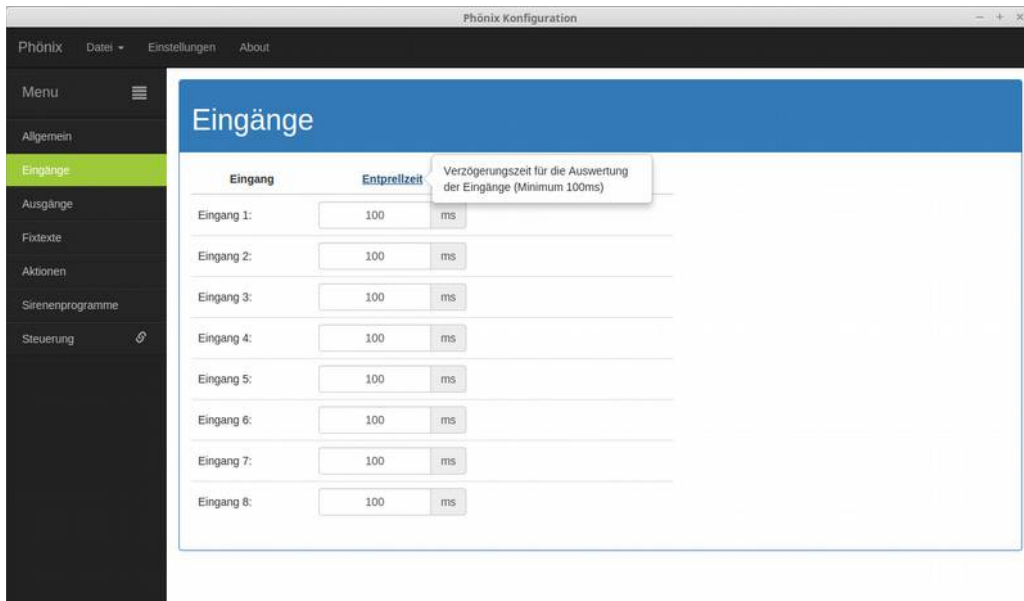


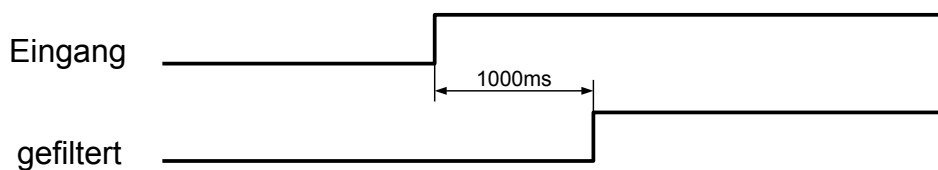
Abbildung 5: Eingänge

Die 8 potentialfreien Eingänge können in ihrer Empfindlichkeit unabhängig konfiguriert werden. Die konfigurierbare Entprellzeit wird als Parameter für einen digitalen IIR Filter verwendet.

### Beispiel:

Eingang wird mit einer Entprellzeit von 1000ms konfiguriert.

Der Eingang wechselt seinen Zustand (z.B. Taste gedrückt) und bleibt konstant auf diesem Zustand. Der Eingang wird entsprechend gefiltert und nach ca. 1000ms erkennt die Software auch einen Zustandswechsel und startet die entsprechenden Aktionen.



Dadurch können zum Beispiel Eingangsänderungen, die durch Störsignale verursacht werden, ausgefiltert werden.

### 4.3 Ausgänge

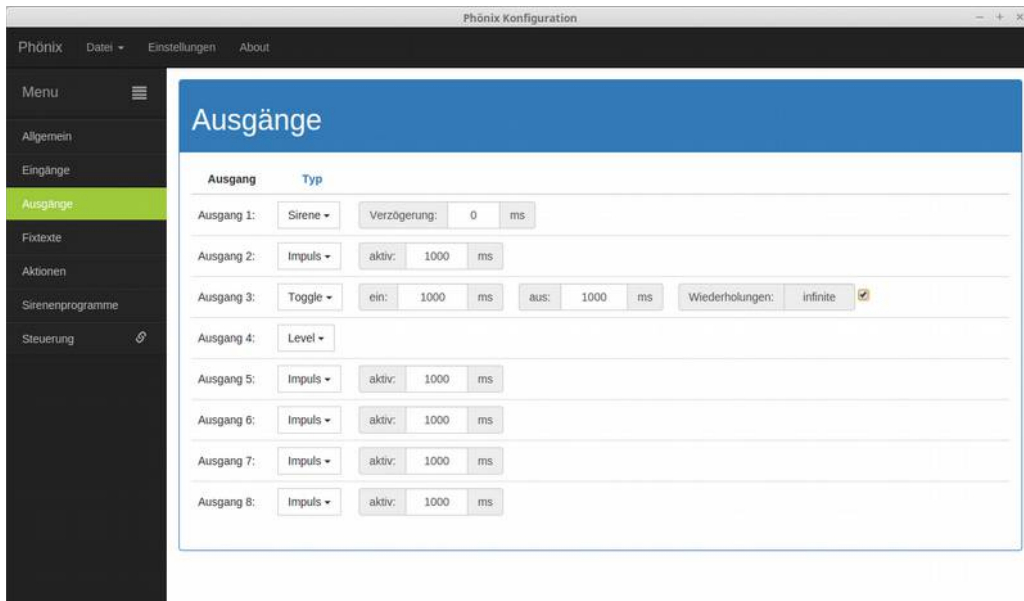


Abbildung 6: Ausgänge

Die Ausgänge können in verschiedene Betriebsarten konfiguriert werden. Ausgang 1 ist standardmäßig als Sirenenausgang konfiguriert, da hier auch das Sirenenrelais angeschlossen ist.

Zu beachten ist, dass erst durch die Zuordnung der Ausgänge zu bestimmten Aktionen (siehe Abschnitt 4.5.1) dieser Ausgang auch verwendet wird. Nur die Konfiguration z.B. als Sirenenausgang reicht nicht aus.

#### Betriebsarten

- **Sirene:** Das für ein bestimmtes Programm konfigurierte Sirenensignal (siehe Abschnitt 4.6) wird hier entsprechend ausgegeben.
- **Impuls:** Der Ausgang wird für eine gewisse konfigurierbare Zeit aktiv.
- **Toggle:** Der Ausgang schaltet zwischen aktiv und inaktiv hin und her. Die Verweildauer im jeweiligen Zustand kann mit den Zeiten „ein“ und „aus“ frei gewählt werden. Der Ausgang kann diesen Zustand für eine gewissen Anzahl an Wiederholungen beibehalten oder dies solange durchführen bis er z.B. durch eine weitere Alarmmeldung wieder gestoppt wird (Einsatzmöglichkeit z.B. für die Steuerung eines Blinklichts bei Sturmwarnung).
- **Level:** Der Ausgang wird durch eine Alarmierung aktiv und durch eine andere Alarmierung wieder inaktiv. Welchen Zustand der Ausgang annimmt, wird bei der Konfiguration der Aktionen festgelegt (siehe 4.5.1).



### 4.5.1 Aktion Funk

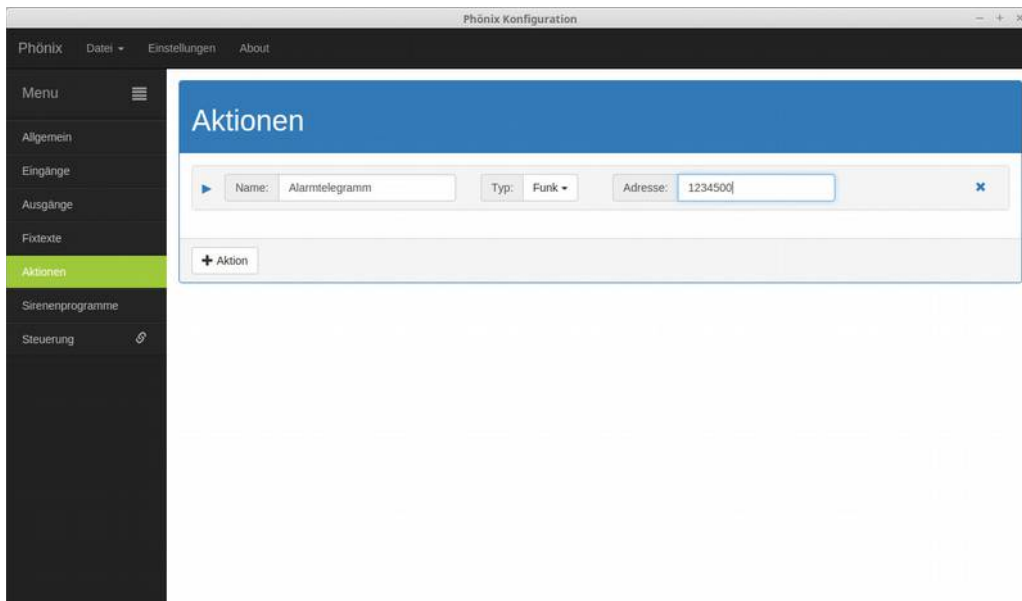


Abbildung 8: Funkalarmierung

Für die Auswertung eines Funktelegramms wird der Typ „Funk“ ausgewählt. Für diesen Typ muss die 7-stellige Auswerteadresse angegeben werden. Mit dem Pfeil vor der Zeile kann die Aktion aufgeklappt werden, um zu definieren was bei diesem Ereignis passiert.

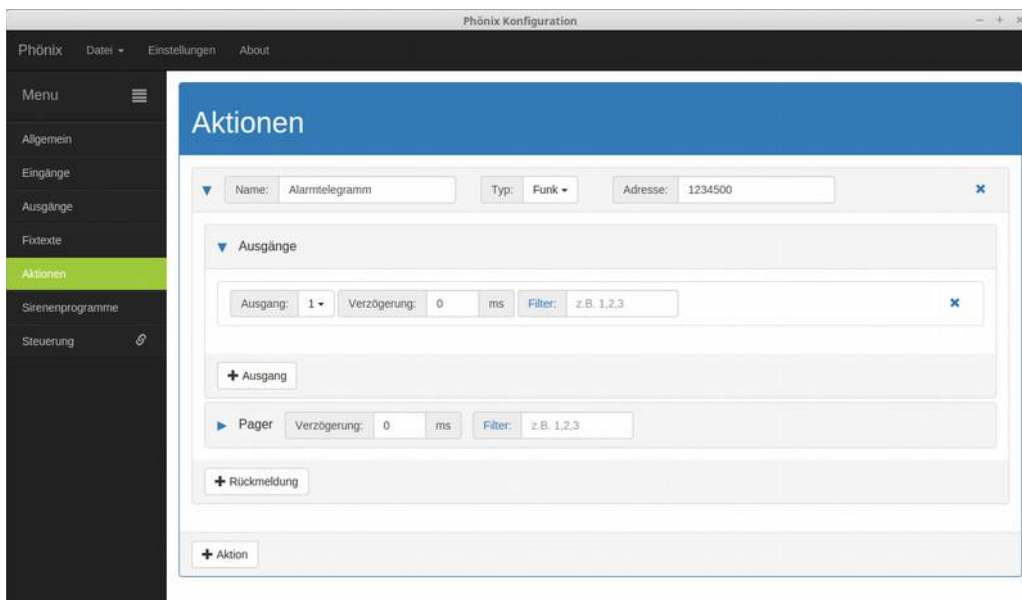


Abbildung 9: Details der Aktion

### Ausgänge

Hier können Ausgänge definiert werden, die bei dieser Aktion entsprechend angesteuert werden. Die Ausgänge verhalten sich entsprechend der vorher festgelegten Betriebsart (siehe Kapitel 4.3).

Wenn der Ausgang in der Betriebsart „Toggle“ konfiguriert ist, muss bei der Aktion noch eingestellt werden, ob der Ausgang „gestartet“ oder „gestoppt“ wird (Abbildung 10).



Abbildung 10: Konfiguration "Toggle" Ausgang

Gleichfalls muss man für einen konfigurierten „Level“ Ausgang bei der Aktion einstellen, ob der Relaisausgang 1 (geschlossen) oder 0 (offen) sein soll (Abbildung 11).



Abbildung 11: Konfiguration "Level" Ausgang

## Verzögerung

Jeder Ausgang kann mit dem Feld „Verzögerung“ noch zeitlich verzögert werden. D.h. nach der Auswertung der Alarmierung kann der Ausgang auch erst nach einer bestimmten Zeit (z.B. 1000ms) aktiv werden.

## Filter

Mit Hilfe des „Filter“ Feldes kann man den Ausgang auch nur für bestimmte Sirenenprogramme definieren. Beispielsweise ist es möglich einen konfigurierten „Toggle“ Ausgang nur für das Programm „Sturmwarnung“ (9) zu starten. Bei allen anderen Programmen jedoch nicht. Dazu muss ins Feld „Filter“ 9 eingetragen werden.

Der selbe Ausgang kann für das Sirenenprogramm „Sturmentwarnung“ (10) wieder gestoppt werden. Die entsprechende Konfiguration ist in Abbildung 12 dargestellt.



Abbildung 12: Filterbeispiel

Will man den Ausgang für mehrere Programme aktivieren, schreibt man alle Nummern mit Beistrich getrennt in das Feld „Filter“.

Beispiel:

1, 2, 3      Bei den Sirenenprogrammen 1, 2 und 3 wird dieser Ausgang aktiv.

## Pager

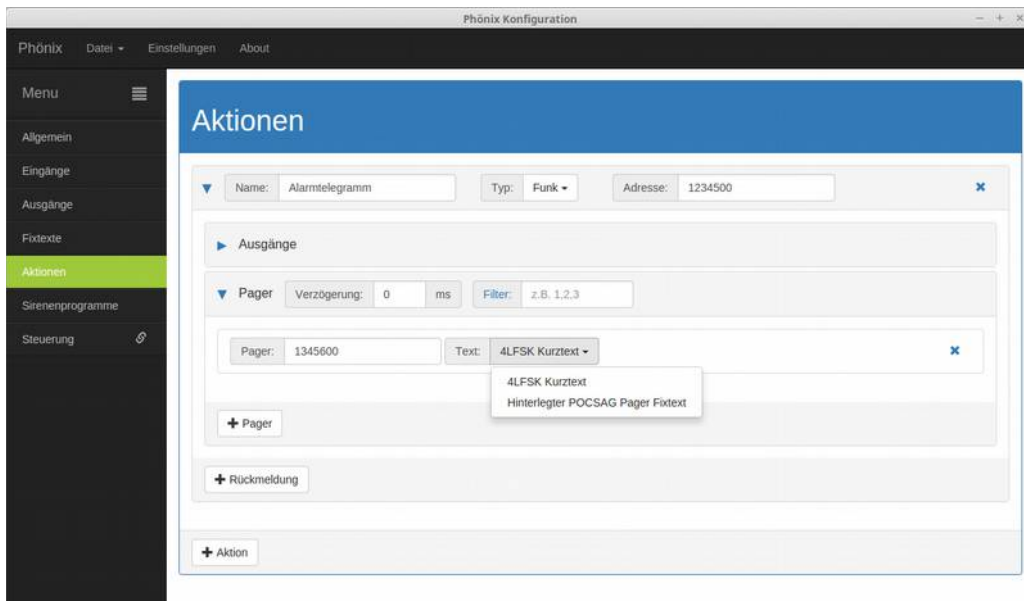


Abbildung 13: Konfiguration Pager

Für eine Aktion können auch ein oder mehrere POCSAG Pager hinterlegt werden. Ein Pager wird mit „+ Pager“ hinzugefügt. Es muss für den Pager die 7-stellige Adresse angegeben werden. Zusätzlich muss der Text ausgewählt werden, der an diesen Pager versendet werden soll. Das Dropdown Menü zeigt alle hinterlegten Fixtexte an, sowie für den Fall der Funkalarmierung die Möglichkeit den empfangenen 4LFSK Kurztext weiterzusenden (Abbildung 13).

Pager lassen sich gleich wie Ausgänge verzögern (Feld „Verzögerung“) und mit dem Filter zu bestimmten Sirenenprogrammen zuordnen.



## Rückmeldung

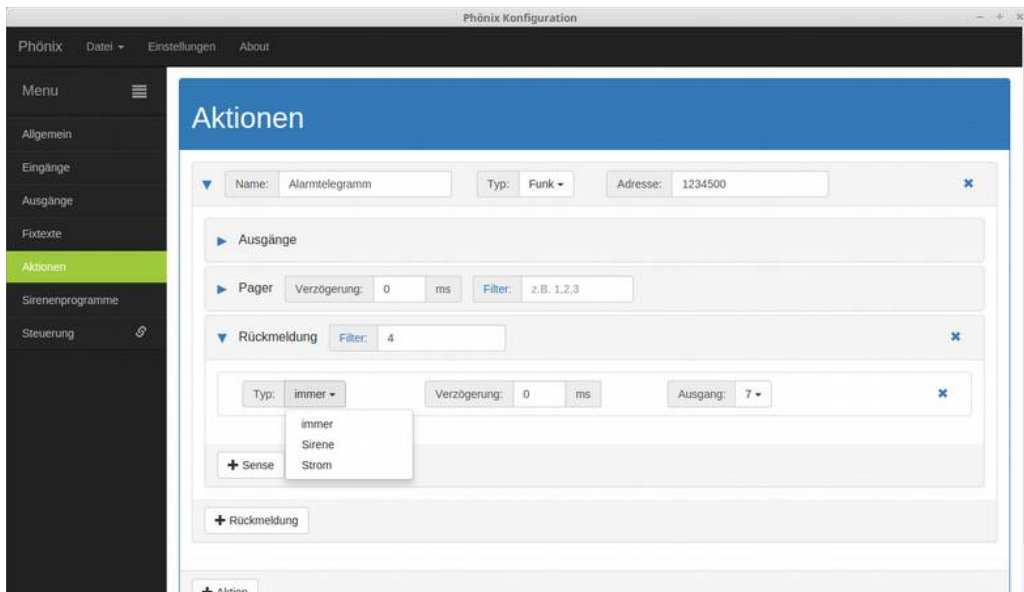


Abbildung 14: Rückmeldung

Weiters kann für eine Aktion eine Rückmeldung konfiguriert werden. Eine Rückmeldung wertet bestimmte konfigurierbare Kriterien (Stromaufnahme Funk, Sirenenrückmeldekontakt) aus und aktiviert bei Erfüllung des Kriteriums nach einer definierten Verzögerung einen Ausgang. Als Ausgang können nur „Impuls“ Ausgänge verwendet werden. Andernfalls wird ein Fehler in der Software angezeigt.

Eine neue Rückmeldung wird mit „+ Rückmeldung“ erstellt. Weiters muss mindestens ein Rückmeldekriterium mit „+ Sense“ hinzugefügt werden. Sind mehrere Kriterien definiert, wird das erste erfüllte Kriterium verwendet. Spätere Kriterien werden nicht mehr berücksichtigt.

Als Rückmeldekriterium stehen folgende Typen zur Auswahl:

- **immer:** der Rückmeldeausgang wird nach der eingestellten Verzögerung immer aktiv.
- **Sirene:** der Rückmeldeausgang wird nach der eingestellten Verzögerung aktiv, wenn die Sirenenrückmeldung erfolgreich war (siehe auch Sirenenrückmeldung 4.1).
- **Strom:** der Rückmeldeausgang wird nach der eingestellten Verzögerung aktiv, wenn die Stromaufnahme des Funkgerätes beim Aussenden der POCSAG Telegramme den eingestellten Schwellwert überschritten hat (siehe auch Rückmeldung Funkgerät 4.1).

Rückmeldungen können mit dem Feld „Filter“ wieder bestimmten Sirenenprogrammen zugeordnet werden.

## 4.5.2 Aktion Eingang

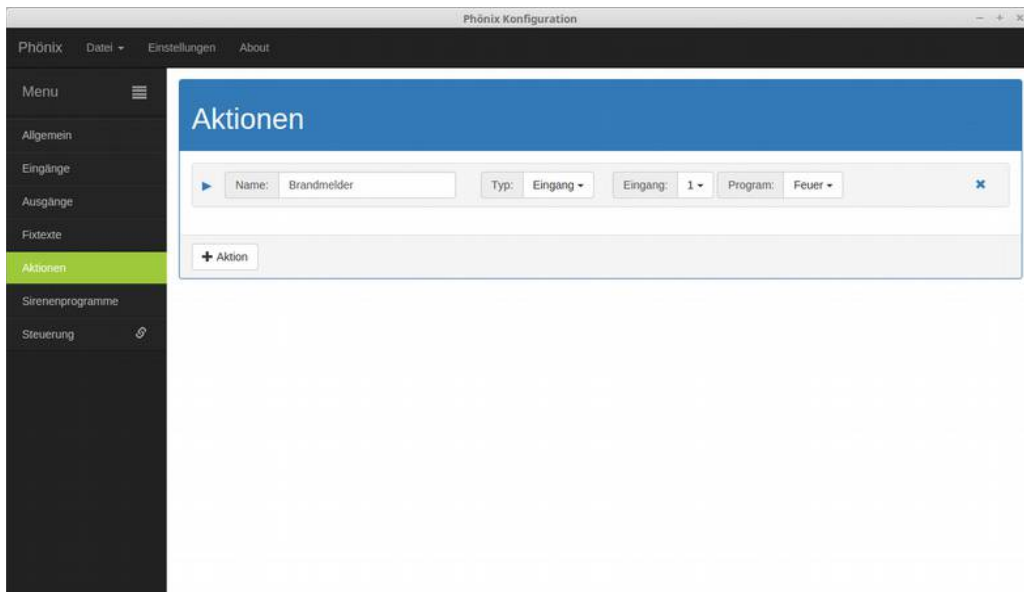


Abbildung 15: Aktion - Eingang

Neben der Funkalarmierung können auch externe Eingänge entsprechende Aktionen auslösen. Hierzu wird als Typ „Eingang“ ausgewählt. Zusätzlich muss die Nummer des Eingangs ausgewählt werden und welches Sirenenprogramm ausgeführt werden soll.

In Abbildung 15 wurde eine Aktion mit dem Namen „Brandmelder“ erstellt. Als Eingang wird Eingang 1 ausgewertet. Wenn dieser Eingang aktiv ist, wird diese Aktion mit dem Programm „Feuer“ ausgeführt.

Wie bei der ausführlich beschriebenen Funkalarmierung (siehe 4.5.1) können auch hier die Ausgänge, Pager und Rückmeldungen entsprechend konfiguriert und zugeordnet werden.

### 4.5.3 Aktion Taste

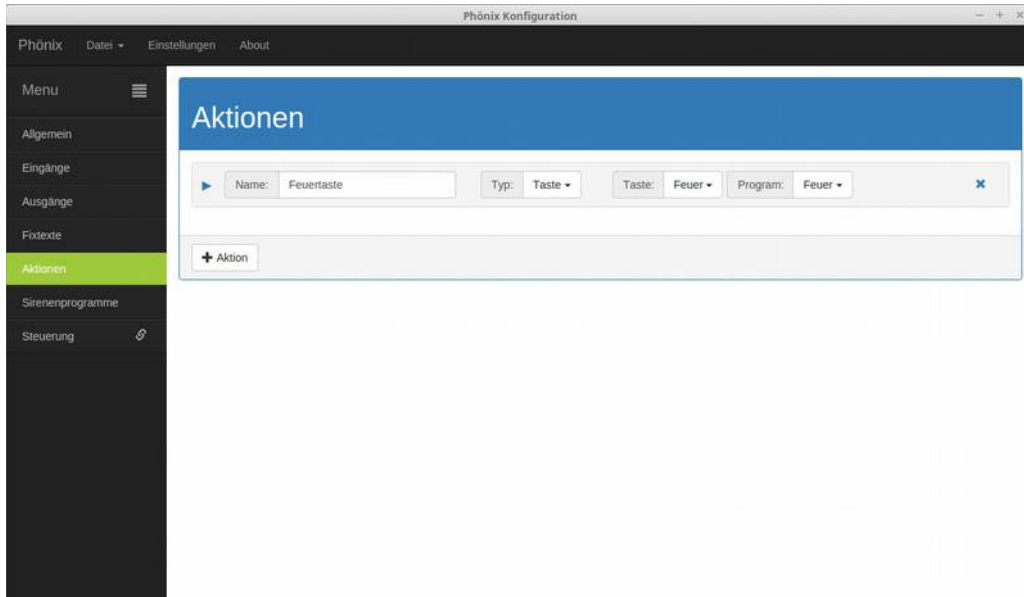


Abbildung 16: Aktion Taste

Zwei der 4 Tasten an der Frontplatte (FEUER und TEST) kann über die Software eine entsprechende Aktion zugeordnet werden. Dazu wird als Typ für die Aktion „Taste“ ausgewählt. Weiters muss die Taste („Feuer“ oder „Test“) und das Sirenenprogramm ausgewählt werden.

Die Zuordnung der Ausgänge, Pager und Rückmeldung ist identisch mit der Beschreibung für die Aktion Funk (Kapitel 4.5.1).

### 4.5.4 Aktion Spannung

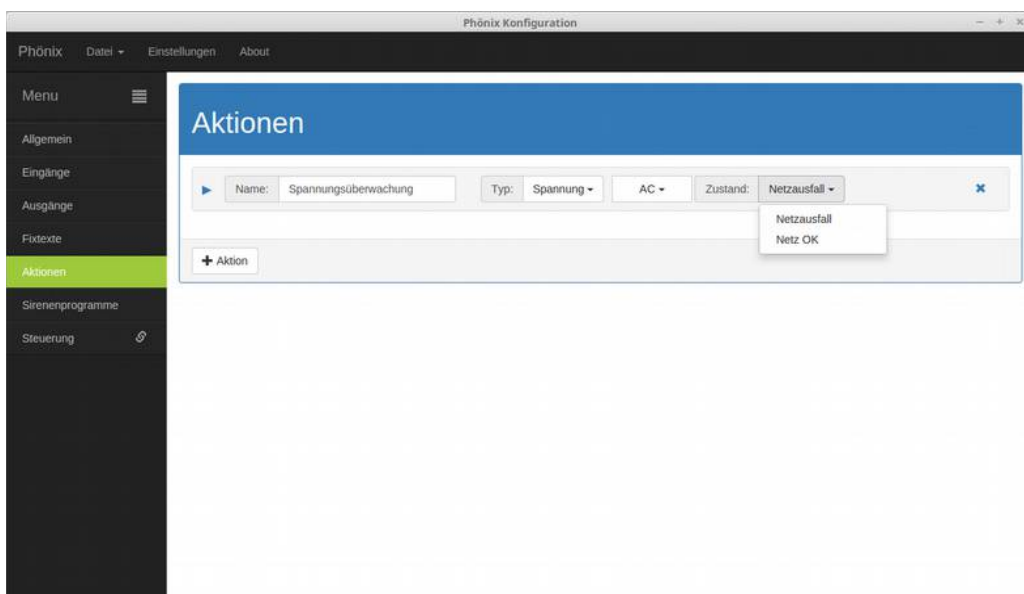


Abbildung 17: Aktion Spannung

Die Sirenensteuerung überwacht die Spannungsversorgung, sodass es möglich ist entsprechende Aktionen für verschiedene Ereignisse zu starten.

Dazu wird als Typ „Spannung“ ausgewählt. Danach können Ereignisse der Netzversorgung (AC) oder der Batterie (Batterie) ausgewertet werden.

- **AC:**
  - **Netzausfall:** Die Aktion wird gestartet, wenn die Netzspannung ausgefallen ist und die Anlage über den Akku versorgt wird.
  - **Netz OK:** Die Aktion wird gestartet, wenn nach einem Netzausfall die Versorgung über die Netzspannung wieder erfolgt.
- **Batterie:**
  - **Unterspannung:** Wenn die Batteriespannung unter den Grenzwert von ca. 11V fällt, dann wird diese Aktion ausgeführt.
  - **Batterie OK:** Steigt die Batteriespannung wieder über den Grenzwert, dann wird diese Aktion gestartet.

#### 4.5.5 Aktion Watchdog

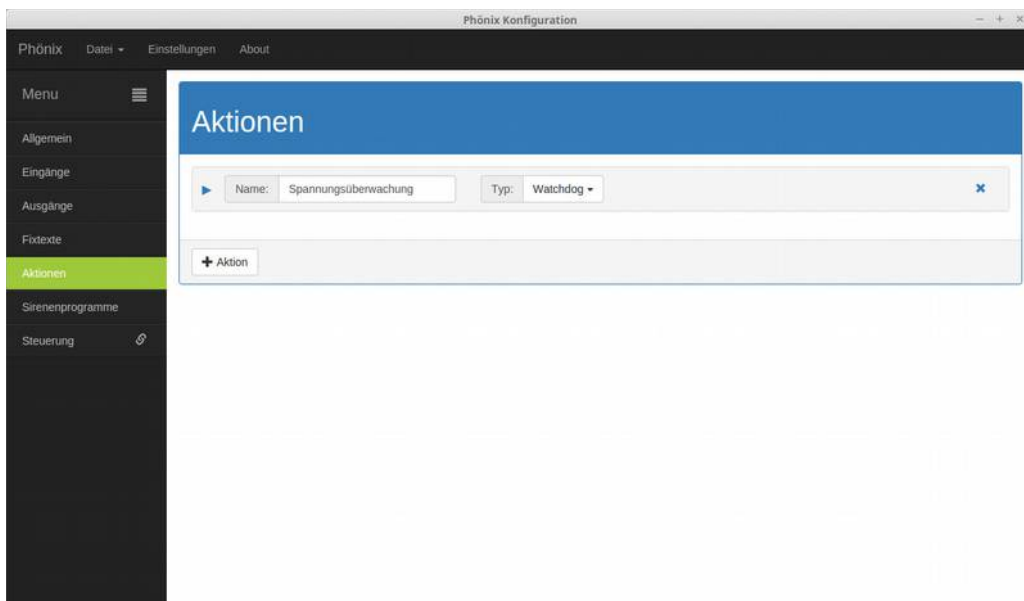


Abbildung 18: Aktion Watchdog

Die Prozessoren auf der Hauptplatine überwachen sich jeweils mit einem Watchdog. Resetiert sich ein Prozessor davon aufgrund eines Watchdog Events, kann dies ebenfalls eine Aktion auslösen.

Gleich wie bei den obigen Events können auch hier Ausgänge aktiviert, Pager alarmiert und Rückmeldungen getriggert werden (Kapitel 4.5.1).

#### 4.5.6 Aktion PC

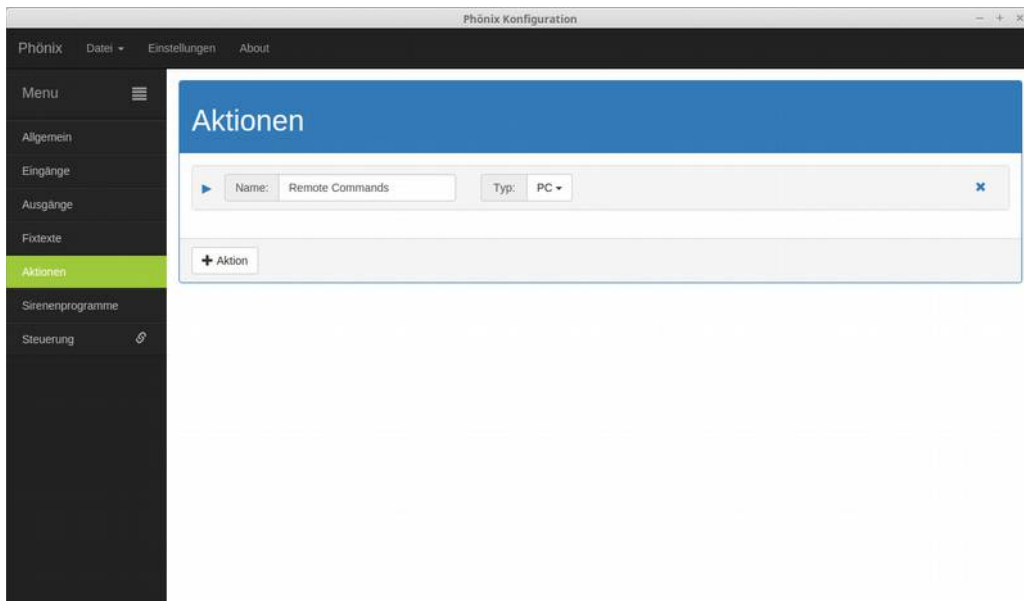


Abbildung 19: Aktion PC

Die Sirenensteuerung kann auch über die PC Software gesteuert (siehe Kapitel 4.8.3) werden. Damit dies funktioniert, muss eine Aktion mit dem Typ „PC“ vorhanden sein. Von der PC Software können Befehle geschickt werden, die z.B. das Sirenenprogramm „Probe“ auslösen.

Damit ein Ausgang entsprechend geschaltet wird, muss auch ein Ausgang dieser Aktion zugeordnet werden. Für die Ausgabe der Sirenensequenz muss allerdings nicht unbedingt der spezielle Sirenenausgang (Ausgang 0) verwendet werden.

## 4.6 Sirenenprogramme

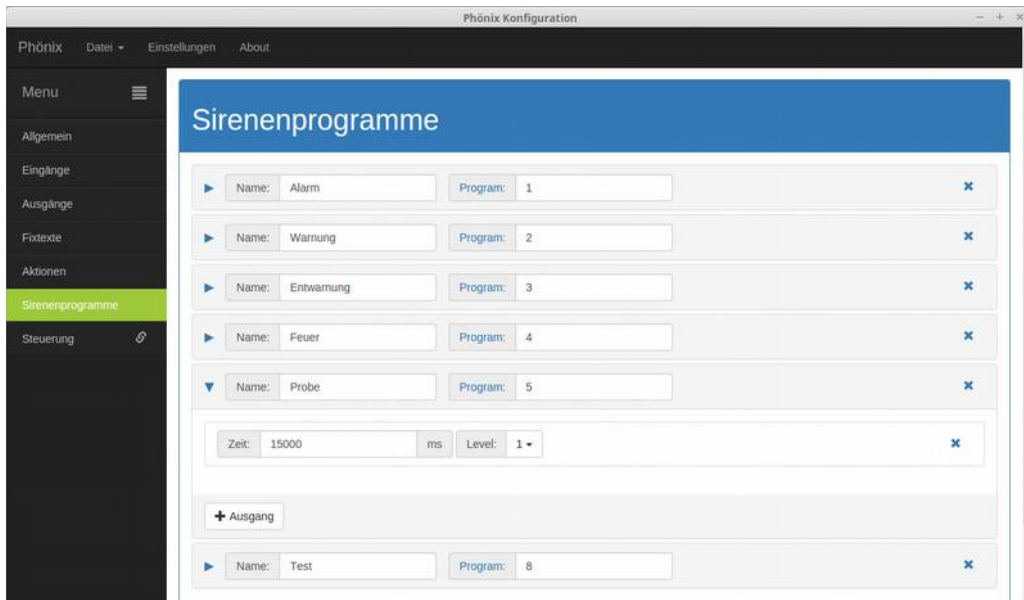


Abbildung 20: Sirenenprogramme

Die Sequenzen für die Sirenenprogramme sind ebenfalls frei definierbar. Bestehende Programme können verändert bzw. neue können angelegt werden. Damit die Alarmierung über Funk funktioniert, muss die Nummer des Sirenenprogramms (Feld „Programm“) mit der im Funkprotokoll definierten Programmnummer übereinstimmen.

Der Zustand (Level: 1 aktiv, 0 inaktiv) und die Dauer dieses Zustands werden entsprechend definiert.

In Abbildung 20 ist dies für das Programm „Probe“ dargestellt. Der Zustand „aktiv“ (1) ist für 15s aktiv.

## 4.7 Einstellungen

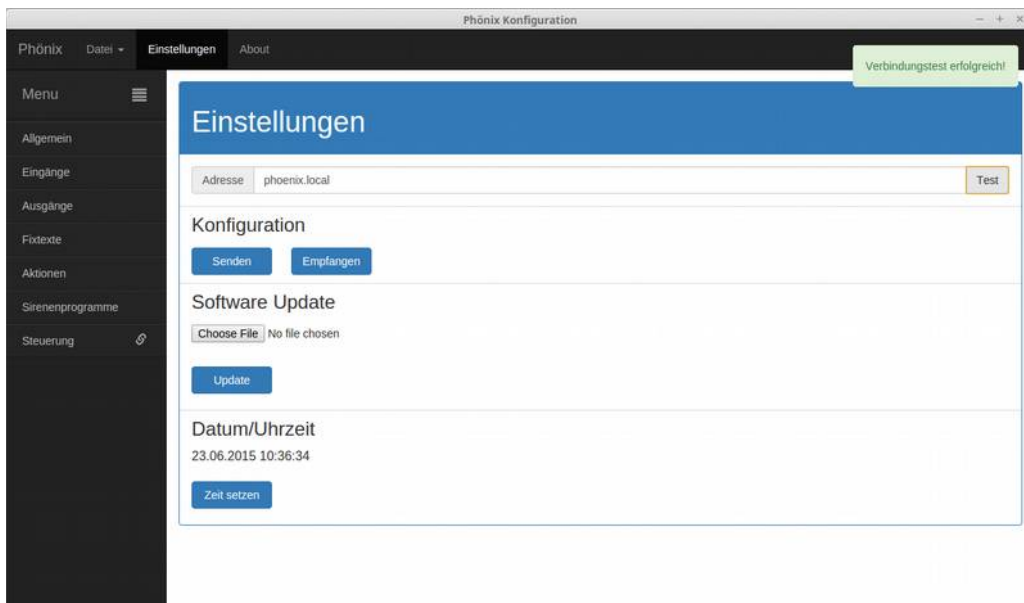


Abbildung 21: Einstellungen

Die Verbindung mit der Sirenensteuerung zur Übertragung der Konfiguration wird ebenfalls über die PC Software hergestellt. Dazu muss über das Einstellungsmenü in der Menüleiste erst die Adresse der Steuerung angegeben werden. Die Adresse lautet

phoenix.local

Diese Adresse wird in das dafür vorgesehene Feld eingetragen. Mit dem Button „Test“ kann die Verbindung überprüft werden. Wenn die Sirenensteuerung über ein Netzkabel mit dem PC verbunden ist, sollte nach kurzer Zeit (unter Windows kann es einige Sekunden dauern) ein grünes Benachrichtigungsfeld rechts oben mit dem Text „*Verbindungstest erfolgreich*“ erscheinen. Kann die Verbindung nicht hergestellt werden, erscheint „*Verbindungsfehler: Überprüfen sie die angegebene Adresse*“.

Unter den Adresseinstellungen befinden sich drei weitere Abschnitte:

- **Konfiguration:**
  - **Senden:** Damit kann die aktuelle Konfiguration an die Steuerung übertragen werden. Ein Benachrichtigungsfeld informiert den User über den Erfolg der Übertragung. Die Übertragung dauert in der Regel nur wenige Sekunden.
  - **Empfangen:** Mit diesem Button wird die aktuelle Konfiguration der Steuerung empfangen. Ein Benachrichtigungsfeld informiert den User über den Erfolg der Übertragung. Die Übertragung dauert in der Regel nur wenige Sekunden.
- **Software Update:**

Für den Fall eines neuen Softwareupdates für die Phönix Sirenensteuerung, kann dieses direkt mit der PC Software eingespielt werden.

  - **Choose File:** Auswahl einer Datei für das Softwareupdate.
  - **Update:** Übertragung des Softwareupdates an die Steuerung.

- **Datum/Uhrzeit:**

Wenn die Steuerung von der Stromversorgung getrennt wird, kann es sein, dass die Echtzeituhr das aktuelle Datum und die Uhrzeit verliert (trotz Backup Batterie).

- **Zeit setzen:** Mit diesem Button kann das aktuelle Datum und die aktuelle Uhrzeit an die Steuerung übertragen werden.

## 4.8 Steuerung

In der linken Navigationsleiste gibt es noch den Menüpunkt „Steuerung“. Darüber lässt sich der aktuelle Status und das Log File der Steuerung auslesen. Außerdem kann die Sirenensteuerung ferngesteuert werden, wenn dies wie in Abschnitt 4.5.6 beschrieben aktiviert wurde.

Damit dieser Menüpunkt sinnvolle Werte liefert, muss eine Verbindung mit einer Phönix Steuerung möglich sein (siehe Adresseinstellungen in Abschnitt 4.7).

### 4.8.1 Logging

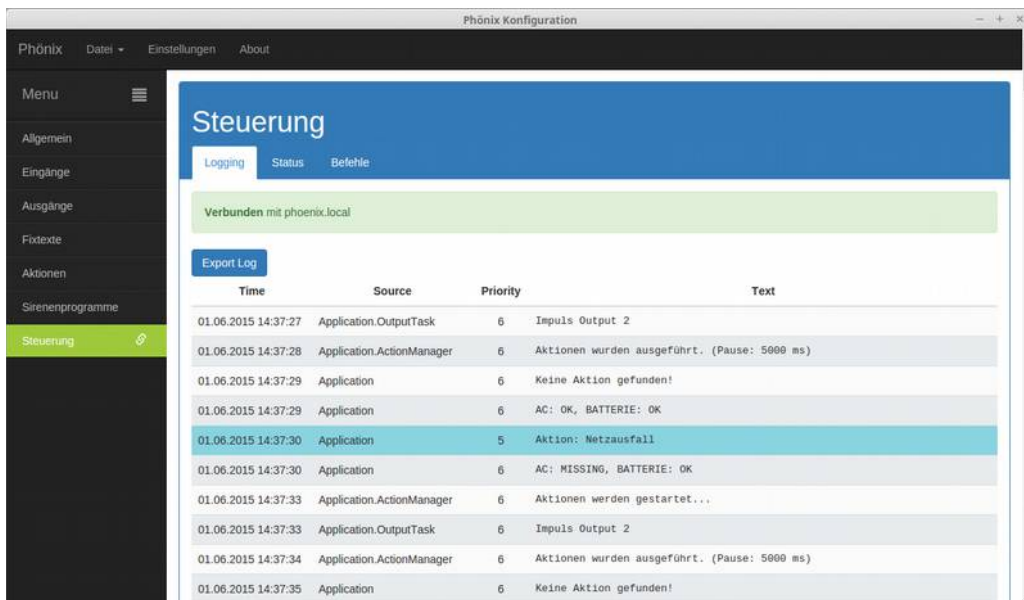


Abbildung 22: Logging

Wenn der Menüpunkt „Steuerung“ aufgerufen wird, erscheint das Logging. Dies wird direkt von der Sirenensteuerung abgerufen. Es werden die letzten 200 gespeicherten Log Einträge abgerufen. Neue Einträge werden in Echtzeit an die Liste angefügt. Das Programm zeigt immer den aktuellsten Eintrag an.

Scrollt man nach oben, gibt es noch einen Button „Export Log“, um das Log in eine CSV Datei zu exportieren. Diese CSV Datei kann in weiterer Folge auch in Excel importiert werden.

Neben dem Logging kann auch der aktuelle „Status“ und die Fernsteuerung über „Befehle“ angezeigt werden.



## 4.8.2 Status

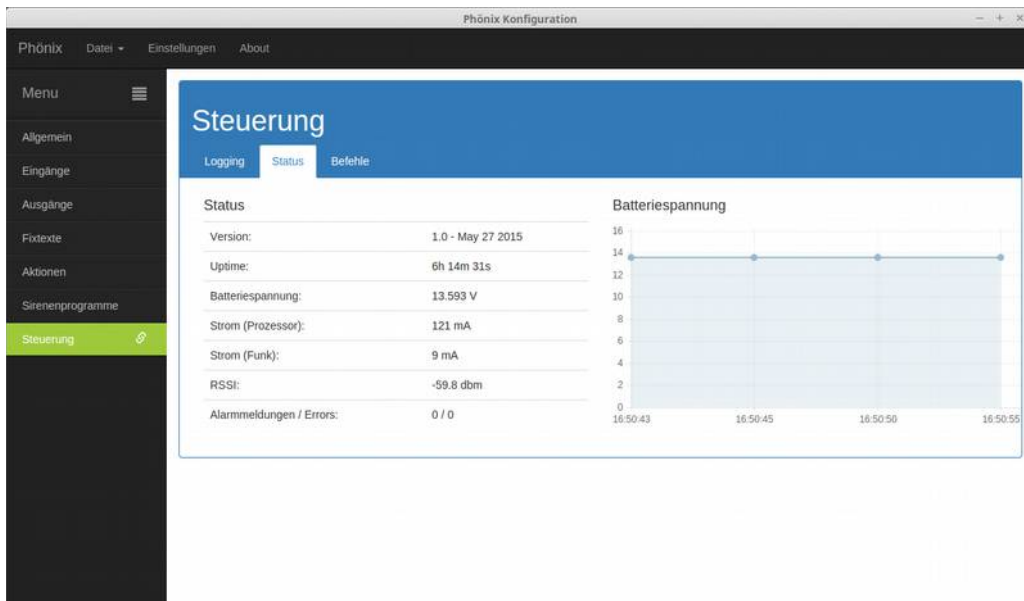


Abbildung 23: Statusanzeige

Unter diesem Menüpunkt findet sich der aktuelle Status der Sirenensteuerung. Der Status wird alle 5s automatisch aktualisiert.

Folgende Eigenschaften werden angezeigt:

- **Version:** aktuelle Software Version
- **Uptime:** Die Zeit wie lange die Steuerung ohne Reset läuft.
- **Batteriespannung:** Aktuelle Batteriespannung. Das Diagramm rechts zeigt den Verlauf der Batteriespannung an.
- **Strom (Prozessor):** Aktueller Strombedarf der CPU Platine.
- **Strom (Funk):** Aktuelle Stromaufnahme des Funkgeräts.
- **RSSI:** Aktueller RSSI Wert, der vom Funkgerät geliefert wird.
- **Alarmmeldungen / Errors:** Anzahl der Alarmmeldungen bzw. der Fehler.

### 4.8.3 Befehle

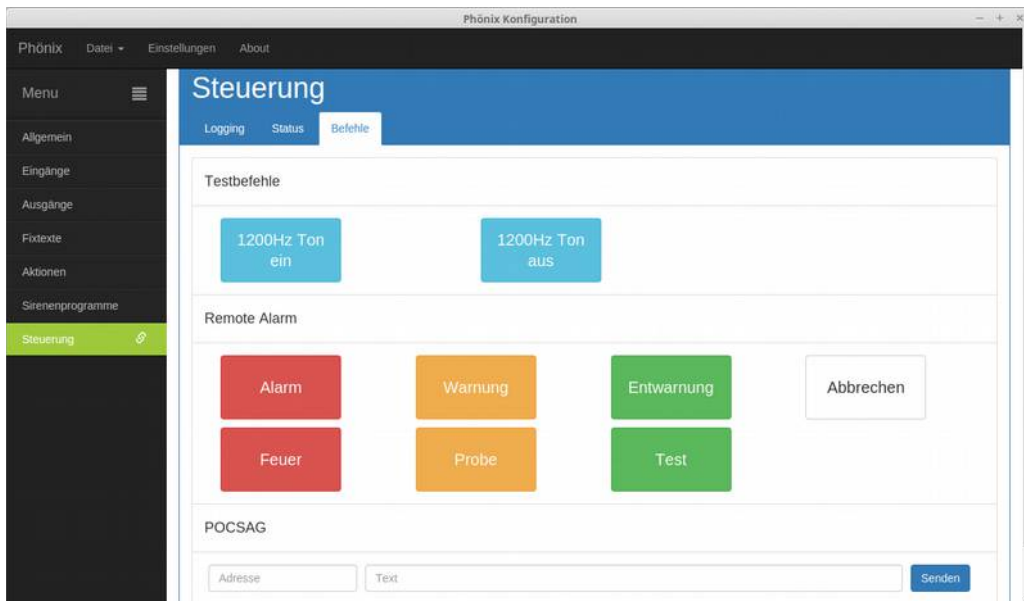


Abbildung 24: Fernsteuerung

Wie bereits weiter oben erwähnt, kann die Phönix Sirenensteuerung über das PC Programm auch ferngesteuert werden.

- **Testbefehle:**
  - **1200 Hz Ton:** Aktiviert bzw. deaktiviert die Generierung eines 1200 Hz Testtons. Gleichzeitig wird auch die Sendetaste des Funkgerätes aktiviert. Damit kann der Sender des Funkgerätes getestet werden.
- **Remote Alarm:**

Für den Remote Alarm muss in den Aktionen eine Aktion mit dem Typ „PC“ vorhanden sein. Wenn diese vorhanden ist, kann mit den Schaltflächen das entsprechende Programm ausgelöst werden. Zusätzlich kann mit „Abbrechen“ eine Ausführung auch abgebrochen werden.
- **POCSAG:**

Der letzte Abschnitt erlaubt einen beliebigen Text an einen beliebigen Pager zu senden. Dazu muss die 7-stellige Adresse des Pagers und der Text in den entsprechenden Feldern angegeben und mit „Senden“ abgeschickt werden.

## **5 Anhang**

### ***5.1 Schaltplan und Lageplan CPU Core Board***

### ***5.2 Schaltplan und Lageplan Backplane***