

# AIIFMS

## Proxy-Server und Mehr für FMS Crusader, FMS32-Pro, POC32 und BosMon

### Kurzanleitung

Stand: 14.03.2011 – Version 1.7.141

#### Einleitung

Das Programm AIIFMS dient als Umsetzer zwischen FMS32-Pro und FMS Crusader. Das heißt, man kann einen Crusader Client mit FMS32pro benutzen oder FMS32pro als Client zum Crusader. Die jeweils aktuelle Version findet sich unter <http://download.arne.de/>

Das Programm ist Freeware. Es gibt die Möglichkeit eine personalisierte Version zu erhalten, die teilweise erweiterte Funktionen besitzt. Da ich jedoch nicht gewerblich Software entwickle, gibt es diese Versionen nicht käuflich zu erwerben. Ich freue mich jedoch über kleine Aufmerksamkeiten, wie z.B. Dinge von meinem [Amazon Wunschzettel](#), oder andere Kleinigkeiten. Nicht der Wert, sondern der gute Wille zählt.

#### Installation

Die Dateien in ein Verzeichnis entpacken und AIIFMS.exe starten. Falls es Fehlermeldung über fehlende DLLs oder OCX Dateien gibt, bitte per E-Mail [allfms1@it-smart.com](mailto:allfms1@it-smart.com) melden, dann stelle ich den kompletten Installer zur Verfügung. Wenn der Webserver bzw. der Google Earth Client genutzt werden soll, dann sollte das Programm das gleiche Verzeichnis installiert werden, indem sich FMS32Pro oder der FMS Crusader befinden.

#### Grundkonfiguration

Es gibt zwei Bereiche: ‚Input‘ und ‚Output‘. Bei Input werden die Server und Ports definiert unter denen die echten FMS32-pro, Crusader oder POC32 Server erreichbar sind. Es können Daten von bis zu vier Servern gleichzeitig entgegengenommen werden, dabei ist es egal ob es sich um FMS32pro oder Crusader oder POC32 handelt – AIIFMS erkennt die Servertypen automatisch. Im ersten Feld neben ‚Server‘ wird die IP Adresse oder der DNS-Name des Servers angegeben. Wenn der Server auf dem gleichen Rechner wie AIIFMS läuft, kann man hier 127.0.0.1 angeben. Im nächsten Feld wird der Port eingetragen. Die Standardeinstellungen für Crusader sind 7778, für FMS32pro 9300 und für POC32 8000.

Falls es sich beim Server ebenfalls um AIIFMS oder den FMS-Crusader handelt, so kann dieser die Verbindung mit einem Passwort schützen. Dieses Passwort muß dann im AIIFMS Input ebenfalls eingetragen werden.

Beim aktivieren von ‚UDP Port‘ hört AIIFMS auf UDP Pakete in einem bestimmten Format. Dies stammt aus Zeiten in denen FMS32pro noch keine Server Funktionalität hatte und ist seit Version 3.2.1 nicht mehr nötig und wird daher in dieser Anleitung nicht näher beschrieben.

Unter ‚Output‘ werden die Server TCP-Ports konfiguriert auf denen AIIFMS die Daten zur Verfügung stellt. Falls Crusader/FMS32pro auf dem selben Rechner laufen, müssen hier natürlich andere Ports als bei ‚Input‘ definiert werden.

Ab AIIFMS 1.3 kann die Verbindung mit einem Passwort geschützt werden. D.h. der AIIFMS Server sendet nur Daten an den Client, wenn dieser beim Verbindungsaufbau das korrekte Passwort übertragen hat. Derzeit wird die Passwort Funktionalität nur vom FMS-Crusader, dem MiniClient und natürlich AIIFMS unterstützt.

Ab Version 1.7 ist die Unterstützung für [BosMon](#) als Auswerter implementiert. Zur Verbindung gibt es zwei Möglichkeiten. A) über die BosMon Ereignisse. Dort ist die Adresse des AIIFMS Servers unter ‚An Alarmierungstool senden‘ anzugeben sowie ein TCP Port für die Verbindung. Im AIIFMS ist dieser Port im Feld ‚BosMon‘ einzustellen und der entsprechende Haken zu setzen. Der Nachteil dieser Methode ist, dass sie für jedes Telegramm eine neue Verbindung aufbaut und daher relativ viel Last erzeugt (also eher nur für Alarmierungen zu verwenden, nicht für FMS Telegramme). B) Über den BosMon Webserver. Da BosMon mehrere ‚Kanäle‘ zur Verfügung stellen kann, ist der gewünschte in der Server-Adresse bei AIIFMS mit anzugeben. D.h. man gibt bei Server1-Server4 den DNS-Namen oder die IP Adresse des BosMon Servers an, gefolgt von einem Schrägstrich (/) und dem Kanal. Z.B. „127.0.0.1/FMSKanal“. Bei Port dann den TCP Port des BosMon Webserver. Bei „Paßwort“ ist der

BosMon Benutzer und Paßwort durch Doppelpunkt (:) getrennt anzugeben. Z.B. „Arne:Me1nPa55wort“. In BosMon muss die Option „Senden von Telegrammen aktivieren“ (unter Telegramme/Netzwerk/Senden) eingeschaltet sein.

In neueren Versionen von BosMon können zusätzlich zum Kanal noch Parameter, abgetrennt durch ein ‚?‘ angegeben werden. Für die Parameterübergabe mittels AIFMS sind diese Parameter durch einen weiteren Schrägstrich getrennt anzugeben, das ‚?‘ wird dabei weggelassen.

Beispiel: 127.0.0.1/FMSKanal/num=10&descriptions=short

### **Weitere Konfigurationen**

AIFMS kann bei der Umsetzung bestimmte Status ausfiltern (z.B. Quittungen von Fahrzeugen). Hierzu werden im Feld ‚Unterdrücken‘ die unerwünschten Status in einem dreistelligen Format, durch Semikolon getrennt, angegeben. Die erste Ziffer ist die Richtung (0=FZG->LST, 1=LST->FZG), die nächsten beiden der Status. So ist ein Sprechwunsch (Status 5) z.B. durch 005 darzustellen. In der Regel ist es sinnvoll die Status 014, 114 und 115 zu unterdrücken. Crusader-Initialtelegramme werden mit dem Stichwort ‚IT‘ unterdrückt.

DME und FME Alarmierungen können mittels Konfiguration in der AIFMS.cfg ausgefiltert werden. Hierzu muß der Eintrag ‚FilterDME‘ bzw. ‚FilterFME‘ gefolgt von einem Leerzeichen und der oder den zu unterdrückenden Schleifen/RICs hinzugefügt werden.

Zusätzlich gibt es ab Version 1.3.78 erweiterte Filtermöglichkeiten. Hierzu müssen die Dateien ‚FilterFMS.txt‘, ‚FilterFME.txt‘ und ‚FilterDME.txt‘ editiert werden. Eine kurze Beschreibung des Filterformates findet sich in den jeweiligen Beispieldateien. Existieren die Dateien nicht, oder enthalten sie keine gültigen Einträge, so wird nicht gefiltert. Diese Filterdateien werden erst nach den ‚Unterdrücken‘ Optionen angewendet. D.h. wenn z.B. generell Status 5 unterdrückt wird, dann kann er auch durch ein erlauben (+) in der FilterFMS.txt nicht mehr dargestellt werden.

Ab Version 1.6.121 können die Ausgaben des HTTP-Servers (nur für FMS-Telegramme) zusätzlich gefiltert werden, z.B. um die GPS-Positionen der Fahrzeuge nicht über HTTP auszugeben. Hierzu ist die Datei FilterWeb.txt zu editieren. Das Format ist identisch mit der FilterFMS.txt

Doppelte Aussendungen eines Status oder einer Alarmierung können im nächsten Punkt unterdrückt werden. Hierzu ist der Zeitraum in Sekunden anzugeben innerhalb dessen ein doppelter Status als Wiederholung betrachtet wird. 0 bedeutet keine Unterdrückung.

Der Punkt ‚Initialtelegramme senden‘ definiert wie viele Initialtelegramme (max. 500) ein Client geschickt bekommt. Diese vom Crusader bekannte Funktion funktioniert auch mit FMS32pro Clients – hier werden die Telegramme im normalen Format gesendet.

Die Funktionen ‚Initialstatus‘ und ‚InitialGPS‘: Wenn dort Haken gesetzt sind, schickt AIFMS die letzten empfangenen Status aller Fahrzeug bzw. die letzten GPS Positionen (Status 10) an jeden Client, der sich neu verbindet. Bei Crusader Clients im Format Initialtelegramm beim FMS32pro Client als normales Telegramm. Zu beachten ist, daß AIFMS nur Status schickt, die es auch empfangen hat, d.h. es wird keine Fahrzeug.DAT/Fahrzeugmelder.CRU Datei eingelesen, sondern nur gesammelt. Auch Fahrzeuge, die in den Clients nicht angelegt sind oder Fehlauswertungen. Diese zu ignorieren ist Sache der Clients.

AIFMS unterstützt die Funktion ‚Beim Programmstart vom Server abrufen‘ von FMS32pro für die Telegrammliste und die 5-Ton- und POCSAG-Liste. Allerdings sendet AIFMS höchstens die letzten 500 Ereignisse (Status und Alarmierungen) – d.h. auch wenn z.B. ‚der letzten 96 Stunden‘ angewählt ist kann es sein, dass die 500 im Speicher nur wenige Stunden zurückreichen. Die Ortskennungen werden hierbei nicht im Klartext angezeigt, da AIFMS diese derzeit noch nicht einliest.

Beim Minimieren verschwindet AIFMS im Systray. Ist dies unerwünscht, so muß in die AIFMS.cfg die Zeile ‚Systray 0‘ eingefügt werden.

Die Konfiguration wird automatisch beim beenden von AIFMS gespeichert. Sollte man sich einmal ‚verkonfiguriert‘ haben, oder möchte eine aktualisierte Fahrzeugliste für den Webserver einlesen, kann man im laufenden Betrieb die komplette Konfiguration neu laden, in dem man die SHIFT Taste gedrückt hält und mit der *rechten* Maustaste auf den BEENDEN Knopf klickt.

### **Serielle Schnittstelle**

Mit AIFMS ist es möglich, als ‚Input‘ auch eine über die serielle Schnittstelle angeschlossene Digibox oder einen Funktronic Major 5 FMS zu nutzen. Diese Funktionalität ist nur in der registrierten Version verfügbar. Da für die Unterstützung der seriellen Schnittstelle eine bestimmte DLL installiert sein muß, wird es auf absehbare Zeit zwei Versionen von AIFMS zu download geben - mit und ohne serielle Unterstützung. Die Version ohne serielle Eingabe hat den Vorteil, dass sie auf sehr vielen Systemen direkt läuft, ohne über ein Installationspaket installiert worden zu sein.

Derzeit ist die serielle Anbindung and Motorola CAS 100 in Arbeit.

## **Aktionen Ausführen**

AlIFMS kann beim Eingang von bestimmten Status oder Alarmierungen externe Prozesse ausführen. Hierzu sind diese in der Datei Aktionen.txt zu definieren. Eine kurze Beschreibung findet sich am Anfang der mitgelieferten Bespieldatei.

## **HTTP Server**

AlIFMS hat einen eingebauten, einfachen, Webserver. Somit kann man die Telegramme mit jedem Web-Browser anzeigen. Getestet wurden Firefox und Internet Explorer. Eine Besonderheit ist die Live-Anzeige: Die HTTP-Verbindung wird offen gehalten und eingehende Telegramme SOFORT im Browser angezeigt, d.h. ein Refresh ist nicht nötig. Voraussetzung dafür ist aktiviertes Javascript. Da die HTTP-Verbindung offen gehalten werden muß, kann es sein dass diese Funktion nicht über http Proxy-Server hinweg funktioniert (außerdem scheint der Internet-Explorer Version 7 damit auch Probleme zu haben). Optional kann in der AlIFMS.cfg durch den Parameter ‚HTTPRefresh X‘ ein reload durch den Browser alle X Sekunden angestoßen werden. Dieses Feature ist allerdings noch in der Testphase und scheint nicht immer zuverlässig zu funktionieren.

Damit der AlIFMS Webserver die Fahrzeugnamen und Melder im Klartext ausgeben kann, muß er die Konfigurationsdateien von Crusader oder FMS32pro einlesen können. Die Dateien müssen sich dazu im selben Verzeichnis wie AlIFMS befinden. Es ist daher zu empfehlen, AlIFMS.exe in den Crusader oder FMS32pro Ordner zu kopieren und von dort zu starten. Wenn die Konfigurationsdateien eingelesen wurden, wird das im Statusfenster (oberes Textfenster) angezeigt. Die Fahrzeugübersicht funktioniert derzeit nur mit FMS32pro. Da AlIFMS einige Einstellungen von FMS32pro aus der Registry lesen muß, sollte AlIFMS auf dem gleichen Rechner wie FMS32pro laufen.

Die Statusfarben sind im Programm vorgegeben, können aber in der AlIFMS.cfg angepasst werden. Das Stichwort ‚Farbe‘ wird gefolgt vom Status und der Richtungskennung (0 der 1) und danach die Farbe im HTML Format (#RRGGBB).

Damit GPS Koordinaten der Fahrzeuge korrekt angezeigt werden können (sofern diese gesendet werden) , müssen die Offsets für Länge und Breite definiert werden. Hierzu steht in der AlIFMS.cfg die Zeilen ‚OffsetL1‘ gefolgt vom ersten Offset Wert für die Länge. Gleiches gilt für OffsetL2, OffsetB1 und OffsetB2. Die Bedeutung dieses Offsets ist in der Anleitung zum FMS Crusader recht gut erklärt.

Siehe <http://www.fmscrusader.de/crusader/download.php>

Default Werte sind hier 2 und 1 für die Länge bzw. 24 und 23 für die Breite.

Ab Version 1.3.85 kann die URL für den HTTP-Link zur Kartendarstellung angepasst werden. In der AlIFMS.cfg sind Beispiele für Mapquest und Google Maps enthalten.

Der Webserver kann auch mit Benutzername+Passwort gesichert werden. Hierzu ist in die AlIFMS.cfg für jeden Benutzer die Zeile:

User Name:Passwort

Einzufügen. D.h. Name ist durch einen Doppelpunkt vom Passwort getrennt. Es können mehrere Benutzernamen angegeben werden, jeder in einer eigenen Zeile. Wenn es keine Zeile ‚User‘ gibt, ist der Webserver frei zugänglich.

Die Weboberfläche kann durch Modifikation der entsprechende Stylesheets etwas angepasst werden. In der Beispiel-AlIFMS.cfg ist das Prinzip erklärt und die Vorlage enthalten.

In der Freeware Version schaltet sich der HTTP Server nach 1000 empfangenen Telegrammen ab – das betrifft auch die Google Earth Anbindung.

## **Google Earth als GPS Client**

Über den HTTP-Server lassen sich die Positionen der Fahrzeuge an Google Earth übermitteln. Wenn Google Earth installiert ist, muß zuerst die AlIFMS beiliegende .KML Datei Editiert werden. Im Feld <href>http://127.0.0.1:80/KML</href>

Muß die URL des AlIFMS Servers eingetragen werden. Bitte den Zusatz ‚/KML‘ nicht vergessen. Das Refresh Intervall sollte auf einen sinnvollen Wert gesetzt werden. Möglichst nicht kürzer als 5 Sekunden. Da bei jedem Refresh eine Datenmenge von 5k bis 50k übertragen wird, sollte dieses Feature nicht mit schmalbandigen Anbindungen genutzt werden.

Danach einfach die .KML Datei mit Google Earth öffnen.

Auch hier ist es wichtig, dass die Offsets korrekt definiert wurden.

Die Icons und Textfarben können in der Datei ‚GStyles.txt‘ angepasst werden. Hierzu ist ein wenig KML Grundlagenwissen nötig. Das gibt's in mehreren KML Tutorials im Internet.

In der AlIFMS.cfg können einige Parameter für die Google Earth Anzeige konfiguriert werden:

GEMaxTextzeilen      Anzahl der Statuszeilen, die für jedes Fahrzeug maximal angezeigt werden.

Je höher der Wert, desto mehr Daten werden bei jeder GE Aktualisierung übertragen. Default: 10

GEMaxFzgAlter      Zeitraum in Sekunden, nachdem ein Fahrzeug nicht mehr angezeigt wird, wenn es keinen Status sendet. Default: 86400 (1 Tag)

GEMaxZielAlter            Zeit in Sekunden nachdem ein Ziel gelöscht wird. Default: 1800. Wenn ein Fahrzeug Status 7, 8, 1 oder 2 drückt, wird das Ziel automatisch gelöscht. Wenn jedoch Sternverkehr geschaltet ist, könnten ansonsten Ziele zu lange bestehen bleiben.

### **Logging**

Wichtige Events können in ein Logfile geschrieben werden. Hierzu ist im CFG File der Parameter ‚Loglevel n‘ zu setzen. Wobei n eine Zahl >0 sein muß. Loglevel 1 oder höher: Alle Events, die auch im Programm angezeigt werden (Server gestartet, beendet, etc). Loglevel 3 zusätzlich fehlgeschlagene http authentifizierungen. Loglevel 5: zusätzlich erfolgreiche http authentifizierungen. Loglevel 10: zusätzlich alle http requests.

Der Parameter ‚Logfile‘ ermöglicht die Definition des Log-Dateinamens. Wird er nicht angegeben, so wird ‚AllFMS.log‘ verwendet.

### **Offene Punkte**

Noch nicht alle Funktionen sind vollständig integriert. Ich freue mich immer über Ideen und Verbesserungsvorschläge.

Folgende Punkte sind für zukünftige Versionen geplant:

- Unterstützung der Crusader Fahrzeugübersichten im http Server
- Evtl Anzeige von Alarmierungen in Google Earth
- Weitere Konfigurationsmöglichkeiten, z.B. eigene Definition der Statusklartexte