

Kommunikationsprotokoll für Monitor 2.x

Version 0.1, J. Schmidt vom 23.4.2007

Update auf Version 0.2, Andreas Stolzenberger, 6.12.07

Update auf Version 0.3, Andreas Stolzenberger, 9.1.08

-- DRAFT --

Einleitung

Das Protokoll dient zur Kommunikation zwischen dem monitor-Serverdienst und mehreren per TCP angebotenen Clients. Das Protokoll überträgt nur Textinformationen ohne Sonderzeichen. Gültige Zeichen sind 0-9 und A-F. Der ":" dient als Feldtrenner. Eine Zeile schliesst mit CR-LF ab. Texte werden als ASCII-8-Bit-Zeichen kodiert z.B. "Ärger3" wird zu "C47267657233".

Jede Zeile beginnt mit einem Befehl, der als dreistellige Hex-Zahl definiert wird. dem Befehl folgen Parameter, welche durch ":" getrennt werden. Die Befehle teilen sich in mehrere Gruppen zu verschiedenen Funktionen auf.

Diese Referenz basiert auf dem Protokoll-Design 0.1 vom 23.4.07. Änderungen gegenüber der Version 0.1 sind **rot hervorgehoben**.

Änderungen von Version 0.2 zu 0.3 sind cyan hervorgehoben.

Befehlsgruppen:

100 Steuerbefehle vom Server

200 Steuerbefehle vom Client

300 Daten vom Server zum Client

400 Daten vom Client zum Server

000 Debug-Befehle

Die Befehlsgruppen 800-900 (geplant als Datenbankkommandos) sind in der Protokollversion 0.2 nicht implementiert, da die erste modulare Version von Monitor 2.0 ohne DB-Backend arbeitet.

Liste der Befehle nach Kommunikationsart

1. Login, Versionsabfrage etc.

100 OK vom Server

200 OK vom Client

121,122,111 Gestrichen

101 und 201 Error

Dem Error folgt eine dreistellige Hex-Zahl als Fehlercode. Es gibt folgende Fehler:

000 der „Bullshit“-Fehler. Irgendetwas ist passiert, was nicht hätte passieren dürfen und daher gibt es dafür keine genauere Fehlermeldung.

001 Not logged in: Client versucht Befehle auszuführen, obwohl er nicht angemeldet ist.

002 Not Authorised: Der Client versucht ein Kommando auszuführen, für das seine Berechtigung nicht genügt.

- 003 Login Fehler: Benutzername oder Password falsch
- 004 Protokollfehler: Server oder Client verstehen nicht, was der jeweils andere will.
- 005 Not Implemented: Der Befehl ist bekannt, funktioniert aber (noch) nicht
- 006 Hardware-Error: Der Server hat ein Problem bei der Kommunikation mit der Soundkarte
- 007 Write-Error: Der Server/Client versucht irgendwelche Daten zu schreiben, kann die Datei aber nicht erzeugen oder hat keine Schreibrechte.
- 008 Versionsfehler: Client oder Server kann mit der Protokollversion seines Gegenüber nichts anfangen.
- 009 Funktion deaktiviert: Die angeforderte Funktion steht nicht zur Verfügung.

Inquiry

110 und 210

Client oder Server wollen von Ihrem Partner wissen, wie er heisst und was er so drauf hat.
Beide Kommandos arbeiten ohne Parameter.

111 und 211

Mehrzeilige Antwort auf Inquiry mit drei Parametern:

111/211:Antwort:Text

Der Parameter Antwort kann sein: 1=Programmname, 2=Betriebssystem,3=Programmversion, 4=Protokollversion, 5=Module, 0=Ende.

Dabei darf der Client/Server mehrere Antwortzeilen pro Option senden, z.B. wenn er mehr als ein Protokoll beherrscht.

Der Inquiry ist der einzige Befehl, den beide Seiten ausführen können, ohne dass ein gültiger Login besteht.
Die Antwort wird nicht mit einem weiteren OK quittiert.

Die Versionsnummern werden mit einer Nachkommastelle als vierstellige Ganzzahlen kodiert: Protokoll Version 0.2 wird also zu 0002, Programmversion 2.1 zu 0021.

Anhand der Protokollversion entscheiden Client und Server, ob Sie mit dem jeweils anderen Arbeiten oder nicht.

Beispiel (in allen Beispielen wird der Text zwecks der Lesbarkeit in „“ angegeben, statt in ASCII kodiert):

```
210
111:1:"monitord"
111:2:"LINUX"
111:3:0021
111:4:0010
111:4:0020
111:4:0021
111:5:"REC"
111:5:"MYSQL"
111:0
200
```

```
110
211:1:"Alarmclient"
211:2:"WINDOWS"
```

```
211:3:0815
211:4:0020
211:0
100
```

oder

```
110
211:1:"alter Client"
211:2:"BSD"
211:3:0010
211:4:0001
101:008
```

Login-Prozedur.

220 gefolgt von Username(text), Password(text) und Protokollversion.

Sobald der Link zwischen Client und Server etabliert ist, beginnt der CLIENT die Kommunikation mit dem Inquiry, danach setzt der Server den Inquiry ab. Damit steht fest, mit welcher Protokollversion die Kommunikation zwischen beiden Setein weitergeht.

Nachdem der Client die „100“ vom Server erhalten hat, leitet er den Login ein:

```
220:"user":"password":0020
100
```

oder

```
220:"user":"password":0020
101:003
```

oder

```
220:"user":"password":0010
101:004
```

Logoff

299 und 199 ohne Paramter. Sowohl der Client als auch der Server können die Verbindung trennen. **Die Gegenseite bestätigt den Logoff nicht mit OK sondern ebenfalls mit Logoff.**

```
299
199
```

oder

```
199
299
```

Keepalive

102 und 202 ohne Parameter

Client und Server können nach eigenem ermessen die Gegenstelle fragen, ob die Verbindung noch steht.

Die Antwort lautet entweder, 100/200, Fehler oder gar nix, was auch auf einen Fehler schließen lässt.

Channel Info (in V 0.1 Ping Pong)

203 ohne Parameter

Client erbittet vom Server Informationen zu den verfügbaren Channels.

Der Server antwortet pro Kanal mit einem 103. Die Parameter:

103:Kanalnummer(char):Kanalname(text):Module(binary encoded)

Dabei zählt 103 rückwärts. Daher bedarf es keinen besonderen "End of List" Befehl. **Die 103er Antworten werden nicht mit einem OK quittiert.** Der letzte Parameter gibt die Features an: 2^0=ZVEI 2^1=FMS 2^2=POCSAC 512 2^3 POCSAC 1200

Beispiel:

```
203
103:3:"Feuerwehr":3 {FMS und ZVEI}
200
103:2:"RD":2 {nur FMS}
200
103:1:"Kat":1 {nur ZVEI}
200
103:0:"Betrieb":6 {POCSAC und FMS}
200
```

Mögliche Fehlerausgabe:

```
203
101:001 {not logged in}
```

2. Recording

Der Client kann den Server anweisen, einen Kanal aufzuzeichnen.

204:Kanalnummer(char):Dauer(Sekunden int)

Der Server bestätigt mit

104:Kanalnummer(char):Aufnahmestatus(char):Dateiname der Aufzeichnung **mit Pfadangabe**(text)

Der Aufnahmestatus kann sein: 0 Aufnahme beendet, 1 Aufnahme gestartet oder 2 bereits laufende Aufnahme verlängert.

Das Schema für den Dateinamen der Aufzeichnung legt die *monitord.xml* auf dem Server fest. Default könnte sein: „DatumUhrzeit_IPdesClients.wav“.

Die Übertragung der aufgezeichneten Datei vom Server zum Client ist (vorerst) NICHT Sache des Monitor-Programms. Hier muss der Systembetreiber sich um eine SMB/CIFS oder NFS Freigabe kümmern.

Beispiel:

```
300: irgendeine ZVEI-Alarmierung
204:1:60
104:1:1:"rec/200712122359_192168001005.wav"
```

60 Sekunden Später

```
104:1:0:"rec/200712122359_192168001005.wav"
200
```

oder

```
204:1:60
101:009
```

oder

```
300: irgendeine ZVEI-Alarmierung
204:1:60
104:1:1:"rec/200712122359_192168001005.wav"
```

etwas später

```
300: noch eine ZVEI-Alarmierung
204:1:600
104:1:2:"rec/200712122359_192168001005.wav"
```

und nach zehn Minuten

```
104:1:0:"rec/200712122359_192168001005.wav"
200
```

mit der aktuellen Code-Basis:

```
204:3:120
101:005
```

3. Alarmmeldungen

Server meldet dem Client einen Alarm. Alle Kommandos, egal ob FMS, ZVEI oder POCSAC beginnen mit dem Parameter Zeit/Datum (Sekunden seit 1.1.70), Der Kanalnummer gefolgt von den Alarmdaten. **Das Feld Servername wird gestrichen. Der Client weiss, zu welchem Server er sich verbunden hat.**

Alarmmeldungen vom Server zum Client werden vom Client NICHT mit einem OK bestätigt.

ZVEI:

300:Zeit:Kanalnummer(char):Schleife(text5):Sirenenalarmierung(char):Text

Der Parameter Sirenenalarmierung schlüsselt sich wie folgt auf:

0 ohne Sirene / Melderauslösung, 1 Feueralarm, 2 Probalarm, 3 Zivilschutz, 4 Warnung, 5 Entwarnung.

Das Textfeld kann wegfallen, in diesem Fall bleibt der Doppelpunkt als Zeilenende.

Beispiel

```
300:{zeitangabe}:2:26100:0:
300:{zeitangabe}:2:26250:1:
204:2:180
104:2:1:"200712122359_192168001005.wav"
300:{zeitangabe}:2:26380:1:
300:{zeitangabe}:2:26280:1:
300:{zeitangabe}:2:26560:1:
300:{zeitangabe}:2:26540:1:
300:{zeitangabe}:2:26433:0:
300:{zeitangabe}:2:26104:0:
300:{zeitangabe}:2:26175:0:
204:2:600
104:2:2:"rec/200712122359_192168001005.wav"
```

und nach 10 Minuten

104:2:0:"rec/200712122359_192168001005.wav"
200

FMS

310:{Zeit}:Kanalnummer:FMSKennung>Status:Baustufe:Richtung:TKI:FMSText

Pocsac

320:{Zeit}:Kanalnummer:RIC:Subadresse:Text

Wie bei ZVEI und FMS gilt: Das Textfeld am Ende kan entfallen, dann muss aber ein Doppelpunkt die Zeile beenden.

Kommando: **330 für DTMF entfällt**. DTMF-Codes kommen ohnehin nur mit ZVEI zum Einsatz und sind daher im 300er-Befehl enthalten.

Tabellen

Tabelle 1 : Kommandos

Kommando	Richtung	Funktion	Parameter
100	S -> C	OK	-
200	C -> S	OK	-
101	S -> C	Error	Errocode
201	C -> S	Error	Errocode
110	S -> C	Inquiry (*)	-
210	C -> S	Inquiry (*)	-
111	S -> C	Inquiry Response	Antwort:Wert 1=Name,2=OS, 3=Version, 4=Protokoll, 5=Module,
211	C -> S	Inquiry Response	Antwort:Wert 1=Name,2=OS, 3=Version, 4=Protokoll, 5=Module,
220	C -> S	Login	Username:Password:Protokollversion
199	S -> S	Logout	-
299	C -> C	Logout	-
102	S -> C	Keepalive	-
202	C -> S	Keepalive	-
203	C -> S	Channel Info Request	-
103	S -> C	Channel Info Response	ChannelNumber:ChannelName:Channel:Features
204	C -> S	Record Request	ChannelNumber:Time
104	S -> C	Record Response	ChannelNumber:RecordStatus:Filename
300	S -> C	Alarm ZVEI	Zeit:ChannelNumber:Schleife:DTFM:Text
310	S -> C	FMS	Zeit:ChannelNumber:FMSKennung>Status:Baustufe:Richtung:TKI:F
320	S -> C	Alarm POCSAC	Zeit:Kanalnummer:RIC:Subadresse:Text

(*) Inquiry arbeitet ohne Login

Tabelle 2: Errorcode

Code	Error
000	unbekannter Fehler
001	Not logged In
002	Not Authorized
003	False Login
004	Protocoll Error
005	Not Implemented
006	Hardware Fault
007	Write Fault
008	Version Error
009	Function deactivated