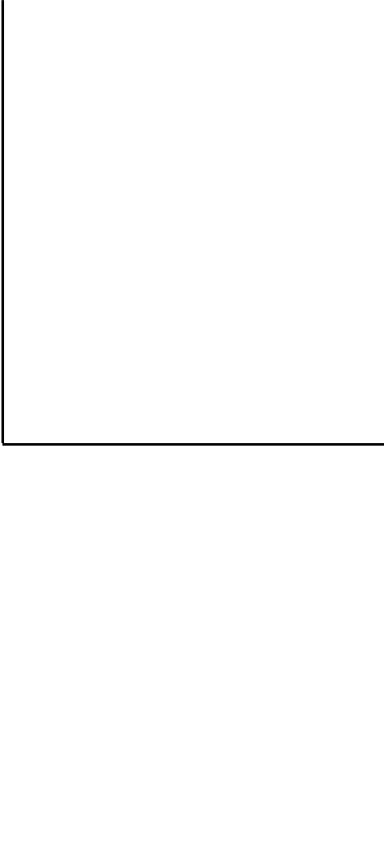


Anleitung

# IO-Mailer



Konfiguration und  
Programmierung

**Rel.** 1.0, Sep. 2000

**Typen:**

57111, 57211

57511, 57611



## Inhalt

1.	Steckverbinder .....	4
2.	Inbetriebnahme .....	5
2.1	IP-Adresse einstellen .....	5
2.2	Gateway und Subnet Mask .....	5
2.3	Vorbereitung zum Mail-Empfang .....	5
3.	Mail-Kommandos .....	7
4.	Konfiguration .....	8
4.1	Wieso XML? .....	8
4.1.1	Unterschiede zu HTML .....	8
4.1.2	Arbeiten mit Dokumentvorlagen .....	9
4.1.3	Die Unterteilung in Hauptzweige .....	11
4.1.4	Zeitangaben .....	12
4.2	Netzwerkumgebung <servers> .....	13
4.3	Postfachzugang <pop-account> .....	14
4.4	Absenderadresse des Mailers <device-id> .....	15
4.5	Erscheinungsbild des Gerätestatus <report> .....	15
4.6	Zeitzone <zone> .....	17
4.7	Zugriffsbeschränkungen <access> .....	17
4.8	Zeitabhängige Steuerung <schedule> .....	19
4.9	Ereignisse und Aktionen <alarm> .....	19
4.9.1	Beispiele .....	21
4.9.2	Einschränkungen .....	24
4.10	Diverse mögliche Probleme .....	24
5.	Anhang: Internetadressen .....	26
5.1	Zum Thema XML .....	26
5.2	Zum Thema Timeserver .....	26
	Technische Daten Typen 57111, 57211 .....	27
	Technische Daten Typen 57511, 57611 .....	28

# 1. Steckverbinder

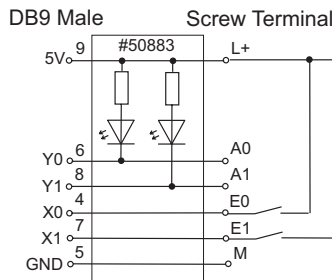
Der serielle Port und die digitalen Ein- und Ausgänge des IO-Mailers liegen an einer DB9 Buchse mit folgender Pinbelegung:

Pin#	Funktion	
1	nc	---
2	TxD	Ausgang
3	RxD	Eingang
4	X0	bin. Eingang
5	GND	0V
6	Y0	bin. Ausgang
7	X1	bin. Eingang
8	Y1	bin. Ausgang
9	VCC	+5V

Die Reihenfolge der seriellen Leitungen RxD/TxD ist hier also gegenüber der DTE-Belegung – wie sie z.B. am COM-Port eines PCs zu finden ist – vertauscht und entspricht damit der DCE-Belegung. Zum provisorischen Anschluß eines seriellen Terminals kann gefahrlos (d.h. ohne zulässige Eingangsspannungen zu überschreiten oder Ausgänge gegeneinander kurzzuschließen) ein 1:1 belegtes seriell Kabel dienen, wie man es auch zum Anschluß eines Modems verwenden würde.

Die binären Eingänge X0 und X1 sind spannungsfest bis +/-30V, ihre Schaltschwelle liegt zwischen +1V und +4V, offene Eingänge werden als binäre 0 gelesen. Y0 und Y1 sind Open-Collector-Ausgänge und können max. +30V und bis zu 100mA schalten. Die Versorgungsspannung VCC kann als Pull-Up-Spannung für die binären Eingänge verwendet werden und darüber hinaus kleine Ausgangslasten bis insgesamt 20mA versorgen.

Für die einfache Anschaltung der Ein- und Ausgänge kann auch der optional verfügbare Schraubklemmadapter Typ 50883 verwendet werden.



## 2. Inbetriebnahme

Für jeden Mailer, den Sie betreiben wollen, benötigen Sie eine *IP-Adresse*, die innerhalb Ihres Netzwerkes eindeutig sein muß. Über DNS sollte ihr ein Hostname zugeordnet sein. Außerdem benötigt der IO-Mailer eine eigene *Mail-Adresse*; auf Ihrem Mail-Server sollte auch ein entsprechendes Postfach neu angelegt werden. („Sollte“ heißt: Zumindest probierhalber ist es auch möglich, den IO-Mailer zu betreiben, ohne daß er Mail empfangen kann oder überhaupt eine gültige Absenderadresse hat. Wenn er aber Mail empfangen und verarbeiten soll, benötigt er auf jeden Fall ein eigenes Postfach und darf auf keinen Fall ein bereits vorhandenes Postfach mitbenutzen!)

Anders gesagt: Um den organisatorischen Aufwand einschätzen zu können, stellen Sie sich den IO-Mailer ungefähr wie einen neuen Mitarbeiter vor. Er bringt als Besonderheit zwar einen eigenen Arbeitsplatzrechner mit, dieser muß aber noch in Ihr Netzwerk und Ihr Mail-System eingebunden werden.

### 2.1 IP-Adresse einstellen

Verbinden Sie sich über ein serielles Terminal (Schnittstellenparameter *9600 Baud, 8 Datenbit, kein Handshake*) mit dem IO-Mailer und schließen Sie dann den Mailer an seine Spannungsversorgung an. Während er startet, tippen Sie am Terminal mindestens dreimal den Buchstaben „x“, bis die Eingabeaufforderung „*IP no.+<ENTER>*“ erscheint. Geben Sie dann die vorgesehene IP-Adresse ein (vier durch Punkte getrennte Zahlen zwischen 0 und 255). Bei Fehlern in der Eingabe antwortet das Gerät mit „*FAIL*“. Schalten Sie in diesem Fall den IO-Mailer wieder aus und beginnen noch einmal von vorn.

### 2.2 Gateway und Subnet Mask

Diese Angaben sind nur erforderlich, wenn nicht alle benötigten Server (vgl. dazu den Abschnitt „Netzwerkumgebung“) im selben Netzsegment wie der IO-Mailer selber zu finden sind. Fragen Sie im Zweifelsfall bitte Ihren Netzwerkadministrator. Um die entsprechenden Daten einzutragen, verbinden Sie sich über Telnet mit dem IO-Mailer (TCP-Portnummer 1111) und wählen Sie das Untermenü „*SETUP System -> Setup TCP/IP*“.

### 2.3 Vorbereitung zum Mail-Empfang

Theoretisch lassen sich alle weiteren Betriebsparameter per E-Mail konfigurieren. Einige dieser Vorgaben müssen allerdings bereits korrekt eingestellt sein, bevor der IO-Mailer überhaupt Mail empfangen kann. Auch dies geschieht über das serielle Terminal – entweder in einem kleinen Frage-und-Antwort-Dialog (wie im folgenden beschrieben) oder über eine komplette Konfigurationsdatei, die Sie per Terminal in den IO-Mailer einspielen (mehr dazu in Abschnitt 4). Wenn ihr Terminal das Senden vorbereiteter Textdateien zuläßt, dürfte dies vermutlich die einfachere Lösung sein.

Verbinden Sie sich also nochmals per Terminal mit dem IO-Mailer (Parameter: *9600 Baud, 8 Bit, Software-Handshake*). Drücken Sie einmal die Leertaste, um das Tatstaturmenü anzuzeigen, und wählen anschließend die Option „*setup (by interview)*“. Die Server für SMTP und POP3 können Sie entweder direkt als IP-Adressen oder über Hostnamen angeben; bei Hostnamen werden Sie aber zusätzlich noch nach der IP-Adresse eines DNS-Servers gefragt. Mit „*Mail-Adresse des Administrators*“ ist die Adresse gemeint, von der aus Sie die übrigen Konfigurationsdaten verschicken wollen. Sie können hier zunächst auch einfach ein „\*“ eingeben, sollten dies in der endgültigen Konfiguration aber unbedingt noch korrigieren (vgl. dazu den Abschnitt „Zugriffsbeschränkungen“).

Um die Konfiguration zu testen, schicken Sie dem IO-Mailer eine Mail mit dem Betreff „*help*“. Wenn alles korrekt eingestellt ist, wird er die Mail innerhalb einer Minute abholen und beantworten. Am seriellen Terminal können Sie diesen Vorgang im Detail beobachten; bei fehlerhaften Einstellungen erhalten Sie dort auch Hinweise auf die mögliche Ursache.

### 3. Mail-Kommandos

Der IO-Mailer reagiert nur auf Mails, deren Betreffzeile mit einem gültigen Schlüsselwort beginnen (wobei Groß-/Kleinschreibung nicht beachtet wird). Gültige Schlüsselwörter sind:

**help**

Liefert (wobei Groß-/Kleinschreibung nicht beachtet wird) eine kurze Zusammenfassung der verfügbaren Kommandos.

**unsubscribe**

Erlaubt den Empfängern automatisch erzeugter Mails, ihre eigene Adresse aus den Konfigurationsdaten zu entfernen. Dies ist eine möglichst unkomplizierte Lösung für den Fall, daß die Einrichtung dieser Benachrichtigungen mit einem Empfänger nicht abgesprochen war. Bei sorgfältig vorbereitetem Einsatz des IO-Mailers sollte das nicht nötig sein.

**resume**

Macht die Wirkung von „*unsubscribe*“ rückgängig.

**status**

Zeigt den Zustand der Ein- und Ausgänge und Zählvariablen an.

**set**

Weist einer oder mehreren Gerätevariablen einen Wert zu. Kann zur Direktsteuerung der binären Ausgänge verwendet werden ( z.B. „*set y0 = 1*“ oder „*set y0=0 y1=1*“), aber auch, um die internen Zähler zurückzusetzen (z.B. „*set n0=0*“).

**info**

Liefert die aktuelle Gerätekonfiguration im XML-Format, das auch zum Einstellen der Konfiguration verwendet wird. Um nur Teile der Konfiguration anzuzeigen, können Sie zusätzlich die Namen eines oder mehrerer der acht Hauptzweige angeben, z.B. „*info alarm*“ oder „*info servers, access*“.

**template**

Hat eine ähnliche Wirkung wie „*info*“; leere Felder in den Konfigurationsdaten werden dabei allerdings ebenfalls ausgegeben. Dies ist zwar weniger übersichtlich, macht es aber leichter, fehlende Informationen zu ergänzen.

**sample**

Liefert eine Beispiel-Konfigurationsdatei mit Triggerbedingungen, durch die bei Anliegen von Signal X0 bzw. X1 jeweils eine Mail-Benachrichtigung erzeugt wird.

**config**

Gerätekonfiguration ändern; die Konfigurationsdaten stehen im Textkörper der Mail. Einzelheiten zum Aufbau der Mail finden Sie in Abschnitt 4.

**update**

Bringt den IO-Mailer in einen Sondermodus für Firmware-Updates. Sie haben anschließend fünf Minuten Zeit, ein TCP-Socket zu Port 8000 des IO-Mailers zu öffnen und ihm eine Textdatei mit der neuen Firmware zu schicken. Vorsicht: Bis das Update erfolgreich ausgeführt wurde (oder bis die fünf Minuten vorüber sind), steht der IO-Mailer für seine normale Funktion nicht zur Verfügung!

Damit ein Mail-Teilnehmer eines dieser Kommandos verwenden kann, muß er in den meisten Fällen ausdrücklich dazu berechtigt sein. Mehr dazu im Abschnitt über Zugriffsbeschränkungen.

## 4. Konfiguration

Der IO-Mailer wird konfiguriert, indem man ihm eine Mail mit dem Betreff „*config*“ schickt, die als Text ein XML-Dokument mit den Konfigurationsdaten enthält. Konfiguration über ein serielles Terminal ist ebenfalls möglich; die zu verwendenden XML-Dokumente sind aber in beiden Fällen gleich.

Die Schnittstellenparameter für das serielle Terminal sind *9600 Baud, 8 Bit, Software-Handshake*. Drücken Sie am Terminal einmal die Leertaste, um das Tastaturmenü anzuzeigen. Wählen Sie dann den Menüpunkt „*configure*“ und übertragen Sie die vorbereiteten Konfigurationsdaten mit der Terminal-Funktion „*Textdatei senden*“.

### 4.1 Wieso XML?

XML (*Extensible Markup Language*) beschreibt Sprachregeln, mit deren Hilfe sich die Konfigurationsdaten für den IO-Mailer übersichtlich in eine Baumstruktur mit benannten Zweigen gliedern lassen. Eine etwas einfachere Syntax (wie z.B. die der *Ini*-Dateien unter Windows) hätte zwar möglicherweise den gleichen Zweck erfüllt, doch XML-Dokumente haben einen großen Vorteil: Es sind derzeit zahlreiche Hilfsmittel erhältlich, die den Umgang mit ihnen erleichtern. Beispielsweise können Sie eine XML-Datei, wie der IO-Mailer sie als Antwort auf das Mail-Kommando „*info*“ schickt, mit dem Internet-Explorer 5 direkt öffnen, betrachten, und ihre Baumstruktur Zweig für Zweig durchblättern.

#### 4.1.1 Unterschiede zu HTML

Falls Sie bereits mit HTML vertraut sind, wird Ihnen an XML vieles bekannt vorkommen. Es gibt aber auch Unterschiede. Die wichtigsten sind:

- Groß-/Kleinschreibung von Element- und Attributnamen wird beachtet.
- Zu jedem öffnenden Strukturelement gehört notwendig auch ein schließendes Element, z.B. `<name>...</name>`. Alleinstehende Elemente sind nur erlaubt, wenn sie durch einen Schrägstrich vor der schließenden spitzen Klammer als solche gekennzeichnet sind: `<name/>`.
- Attributwerte müssen grundsätzlich in Anführungszeichen eingeschlossen sein, z.B. `<name attr="1"/>`
- Es gibt mit Ausnahme der fünf Zeichen `<` `>` `&` `'` `"` (vgl. untenstehende Tabelle) keine Codierungen für Sonderzeichen. Formulierungen wie `&auml;` anstelle von „ä“ sind also *nicht* erlaubt.
- Wie bei HTML gibt es auch bei XML für jedes Dokument ein Strukturelementpaar, das den gesamten Dokumentinhalt einschließt. Anders ausgedrückt: Die Baumstruktur, in die das Dokument gegliedert ist, geht von genau einer Wurzel aus. Doch während diese Wurzel für HTML-Dokumente in jedem Fall `<HTML>` heißt, ist ihr Name bei XML-Dokumenten je nach Anwendung definiert – für die Konfigurationsdaten des IO-Mailers lautet er `<io-mail.1>`.



## 4.1.2 Arbeiten mit Dokumentvorlagen

Der IO-Mailer kann mit dem Mail-Kommando „*template*“ ein Rohgerüst des zur Konfiguration benötigten XML-Dokuments erzeugen. Sie tragen dann lediglich noch Ihre eigenen Konfigurationsdaten ein. Da man den Mailer aber bereits konfiguriert haben müßte, damit er überhaupt ein solches Mail-Kommando verarbeiten kann, steht die gleiche Funktion auch noch einmal am seriellen Terminal zur Verfügung. Wählen Sie dazu im Tastaturmenü den Punkt „*full XML template*“.

Auch wenn Sie bislang weder XML noch HTML kennen und sich nur über die vielen spitzen Klammern wundern, sollten Sie mit dem Ausfüllen dieser Formulare keine größeren Schwierigkeiten haben, solange Sie die folgenden Hinweise beachten:

### Sonderzeichen

In den Texten, die Sie eingeben, dürfen die spitzen Klammern sowie drei weitere Sonderzeichen nicht vorkommen. Ersetzen Sie diese durch Codierungen gemäß der folgenden Tabelle:

<	&lt;
>	&gt;
&	&amp;
"	&quot;
'	&apos;

### Leere Felder ausfüllen

Einfachen Text können Sie zwischen zusammengehörigen öffnenden und schließenden Strukturelementen eintragen, Attributtext zwischen Anführungszeichen. Beispiel:

vorher:

```
<repeat></repeat>
<action var="" val="" />
```

nachher:

```
<repeat>00:00:01</repeat>
<action var="Y1" val="0" />
```

### Lange Texte

Es ist nicht erforderlich, daß öffnendes und schließendes Strukturelement in derselben Zeile stehen, und auch der Text selber darf über mehrere Zeilen verteilt werden. Beispiel:

vorher:

```
<subject></subject>
<message></message>
```

nachher:

```
<subject>
ziemlich lange Betreffzeile, jedenfalls für unsere Verhältnisse
</subject>
```

```
<message>
  Etwas längerer Absatz.
  Zeilenumbruch,
    linker Rand
  und Größe der Wortzwischenräume sind
  bei der Eingabe übrigens egal.
</message>
```

### Löschen und Vervielfältigen

Auch wenn in der ursprünglichen Dokumentvorlage jeder Zweig genau einmal vorkommt: Viele dürfen auch mehrfach verwendet werden, manche können aber auch ganz entfallen. Dabei ist das vom IO-Mailer erzeugte Dokument so formatiert, daß man normalerweise nichts falsch macht, solange man immer nur ganze Zeilen löscht oder vervielfältigt. Beispiel (mehr als einen DNS-Server angeben):

vorher:

```
<dns></dns>
```

nachher:

```
<dns>172.16.232.14</dns>
<dns>172.16.232.15</dns>
```

Noch ein Beispiel, bei dem unbenutzte Elemente und Attribute gelöscht werden:

vorher:

```
<trigger op="DELTA" var="N1" val="3" delay="" recover="" />
<repeat></repeat>
<mailto>test@xyz</mailto>
```

nachher:

```
<trigger op="DELTA" var="N1" val="3"/>
<mailto>test@xyz</mailto>
```

Allerdings: Mit dem Kopieren bzw. Löschen einzelner Zeilen ist es in zwei wichtigen Fällen nicht getan. Erster Fall: Der *alarm*-Zweig darf mehrere *event*-Zweige enthalten, verfügt aber in der Dokumentvorlage zunächst nur über einen einzigen. Wie die korrekte Konfiguration mit mehreren *event*-Zweigen aussieht, ist in den entsprechenden Beispielen des Abschnitts „Ereignisse und Aktionen“ dargestellt.

Der andere Fall hat damit zu tun, daß die Konfiguration in acht unabhängige Hauptzweige unterteilt ist, von denen man gelegentlich einzelne löschen will (mehr dazu im nächsten Abschnitt). Jeder Zweig beginnt und endet mit einer Zeile, die sein öffnendes bzw. schließendes Strukturelement enthält, also z.B. `<access>` und `</access>` für den *access*-Zweig. Es ist nun sowohl erlaubt, nur den Inhalt eines Zweiges (die Strukturelemente bleiben stehen) als auch den kompletten Zweig einschließlich seiner Strukturelemente zu löschen. Beides hat aber unterschiedliche Auswirkungen: Wird nur der Inhalt eines Hauptzweiges gelöscht, so löscht man damit auch alle zugehörigen Daten aus der Gerätekonfiguration (das will man nur in Ausnahmefällen). Löscht man dagegen den Hauptzweig selber, bleibt der entsprechende Abschnitt der Gerätekonfiguration unverändert.

### Kommentare

Der gesamte Text zwischen *den Zeichen* `<!--` und `-->` wird ignoriert. Sie können damit auch ganze Abschnitte Ihres Dokuments quasi ausblenden. Es ist allerdings nicht erlaubt, Kommentare ineinander zu verschachteln. Beispiel:

```
<!-- Hmm... Ist das sinnvoll? -->
<!--
<suspect>*</suspect>
-->
```

## 4.1.3 Die Unterteilung in Hauptzweige

Die Konfiguration ist in acht Hauptzweige unterteilt, die in den folgenden Abschnitten beschrieben werden:

```
<servers>
<pop-account>
<device-id>
<report>
<zone>
<access>
<schedule>
<alarm>
```

Diese Hauptzweige können unabhängig voneinander konfiguriert werden. Das bedeutet, Sie können z.B. die Zeitzone des Mailers ändern, ohne gleichzeitig die Triggerbedingungen neu definieren zu müssen. Aber für jeden Zweig, der beim Verschieken von Konfigurationsdaten überhaupt erwähnt wird, gilt die Regel: Es werden alle vorhandenen Daten gelöscht, bevor neue übernommen werden. Wer also etwa einen neuen DNS-Server eintragen will, muß auch die Time- und Mail-Server kennen und erneut angeben. Ein prinzipielles Hindernis für Änderungen sollte das aber nicht sein, da die aktuellen Konfigurationsdaten jederzeit aus dem IO-Mailer selber abrufbar sind (Mail-Kommandos „*info*“ und „*template*“).

Es ist also empfehlenswert, bei geringfügigen Änderungen wirklich nur die betroffenen Zweige zu ändern. In dem oben erwähnten Beispiel würde man also eine Mail mit dem Betreff „*info servers*“ schicken, das auf diese Weise erhaltene XML-Dokument editieren, den neuen DNS-Server eintragen und dann mit dem Betreff „*config*“ zurückschicken. Man vermeidet damit zwei mögliche Probleme: Erstens stört es unnötig den Betrieb des IO-Mailers, wenn bestimmte Zweige neu geschrieben werden, obwohl sich inhaltlich gar nichts an ihnen ändert (*report*: die Meßwertreihen werden gelöscht, *alarm*: die Ereignisverarbeitung wird neu initialisiert). Und zweitens gibt es Zweige, die einige Mail-Teilnehmer ohnehin nur anzeigen, nicht aber ändern dürfen (*servers*, *pop-account*, *access*).

Die folgenden Beispiele gehen durchweg von dieser Unterteilung in unabhängige Zweige aus und führen jeweils nur die Veränderung eines einzigen Zweigs vor. Beim abgedruckten Code handelt es sich jeweils um komplette XML-Dokumente, die in dieser Form auch direkt vom IO-Mailer akzeptiert würden, als Besonderheit aber eben jeweils nur einen einzigen Hauptzweig enthalten.

Um Mißverständnissen vorzubeugen, soll darum hier einmal angedeutet werden, wie mehrere dieser Beispiele kombiniert werden müßten, wenn mehrere Zweige gleichzeitig geändert werden sollen:

```
<io-mail.1>
  <servers>
  ...
</servers>
<pop-account>
  ...
</pop-account>
</io-mail.1>
```

Das folgende Beispiel dagegen wäre die *falsche* Methode:

```
<io-mail.1>
  <servers>
  ...
</servers>
</io-mail.1>      <!-- So bitte nicht! Alles nach dem -->
<io-mail.1>      <!-- ersten </io-mail.1> wird ignoriert! -->
  <pop-account>
  ...
</pop-account>
</io-mail.1>
```

Oder anders gesagt: Das Strukturelementpaar, welches das gesamte Dokument einschließt (in unserem Fall ist das `<io-mail.1>...</io-mail.1>`), darf naturgemäß nur einmal vorkommen. Andernfalls hätte man es mit mehreren Konfigurationsdateien zu tun – aber Sie dürfen dem IO-Mailer eben immer nur jeweils eine einzige Datei schicken.

#### 4.1.4 Zeitangaben

An mehreren Stellen innerhalb der Konfigurationsdaten kommen Uhrzeiten und Zeitspannen vor. Das hierbei zu verwendende Format ist in allen Fällen (mit Ausnahme der Zeitzone) HH:MM:SS, also z.B. 04:30:00 für halb fünf Uhr früh und 00:00:10 für zehn Sekunden. Zeitspannen können alternativ auch einfach als eine Anzahl Sekunden angegeben werden, z.B. 90 für eineinhalb Minuten oder 0.25 für eine Viertelsekunde.

## 4.2 Netzwerkumgebung <servers>

In Ihrem Netzwerk sollten folgende Server vorhanden sein:

- ein DNS-Server. Es geht auch ohne, aber dann müssen Sie alle anderen Server direkt über ihre IP-Adresse angeben und können nicht auf den einfacher zu handhabenden Hostnamen zurückgreifen.
- ein SMTP-Server und ein POP3-Server zum Senden bzw. Empfangen von Mail. Diese Dienste sind häufig zusammen auf ein- und demselben Server installiert (im Fall des Microsoft Exchange Server handelt es sich sogar um ein- und dasselbe Produkt).
- ein Timeserver nach RFC-868. Dringend empfehlenswert, damit der IO-Mailer korrekte Zeitstempel erzeugen kann. Es gibt weltweit erreichbare Timeserver im Internet (z.B. clock.psu.edu, swisstime.ethz.ch) sowie kostenlose Software, mit der Windows-Rechner als Timeserver eingerichtet werden können (mehr dazu im Anhang). Am einfachsten ist es aber, wenn Sie einen eigenen Unix- oder Linux-Server zur Verfügung haben. Dort brauchen Sie lediglich sicherzustellen, daß in der Datei „inetd.conf“ der als „time“ bezeichnete Dienst aktiviert ist.

Beispiel:

```
<io-mail.1>
  <servers>
    <connect>00:00:05</connect>
    <response>00:01:00</response>
    <terminate>00:00:10</terminate>
    <dns>172.16.232.15</dns>
    <dns>172.16.232.14</dns>
    <smtp>mail.firma.de</smtp>
    <pop3>mail.firma.de</pop3>
    <time>atropos.firma.de</time>
    <time>clock.psu.edu</time>
    <time>time.iem.it</time>
  </servers>
</io-mail.1>
```

Zu jedem Server-Typ können mehrere alternative Adressen bzw. Hostnamen angegeben werden. Dabei versucht der IO-Mailer im Normalfall, sich mit dem jeweils zuerst aufgeführten zu verbinden. Erst wenn das fehlschlägt, geht er der Reihe nach die übrigen Adressen durch.

Mehr als einen DNS- bzw. Time-Server anzugeben, kann auf keinen Fall schaden. Eine Konfiguration mit mehreren SMTP-Servern ist schon eher unüblich, und mehrere POP3-Server zur Auswahl sind normalerweise nicht sinnvoll (aufgrund der Funktionsweise dieses Dienstes: Der POP3-Server repräsentiert eine Sammlung von Postfächern, und wenn er ausfällt, steht deren Inhalt einfach nicht mehr zur Verfügung).

Der Connect-Wert, den Sie hier eintragen, bestimmt, wieviel Zeit der IO-Mailer seinen Servern zum Akzeptieren einer TCP-Verbindung einräumt. Solange nichts anderes angegeben ist, gelten hier fünf Sekunden als Standardwert. Noch größere Werte sind

normalerweise unpraktisch, weil es dann sehr lange dauert, bis der Ausfall eines Servers erkannt wird. Noch folgenschwerer sind dagegen zu kleine Werte, denn es kommt keine Netzwerkverbindungen zustande.

Innerhalb einer bestehenden Verbindungen führt der IO-Mailer mit seinem POP3-Server längere Dialoge. Dabei ist es leider normal, daß stark ausgelastete Server für die einzelnen Antworten sehr viel Zeit brauchen.

Mit dem *Response*-Wert wird eine Obergrenze für diese Antwortzeiten festgelegt. Ist die angegebene Zeit überschritten, beendet der IO-Mailer die Verbindung. Wenn nicht anders angegeben, setzt der IO-Mailer diesen Wert auf eine Minute.

Das Gegenstück zum *Connect*-Wert ist der *Terminate*-Wert, der angibt, binnen welcher Zeit Server einen Verbindungsabbau quitieren müssen. Als Standard gelten zehn Sekunden.

### 4.3 Postfachzugang <pop-account>

Diese Informationen benötigt der IO-Mailer zum Empfang von Mail. Beispiel:

```
<io-mail.1>
  <pop-account>
    <mailbox>eabox1</mailbox>
    <password>caligula</password>
    <interval>00:01:00</interval>
  </pop-account>
</io-mail.1>
```

**Achtung:** Das Postfach, das Sie hier angeben, muß dem Mailer alleine gehören! Denn er löscht sofort alle Mails, die er gelesen hat – auch solche, die er inhaltlich gar nicht versteht. Vermeiden Sie es deshalb unbedingt, das Postfach eines normalen Netzteilnehmers vom Mailer mitbenutzen zu lassen! Auch ein Postfach für mehrere Mailer wäre nicht sinnvoll. (Mails an dieses Postfach würden dann nur jeweils einen der Mailer erreichen; doch welchen, ist nicht vorhersagbar.)

Beachten Sie außerdem, daß der IO-Mailer keine Paßwortverschlüsselung – auch bekannt als MD5- oder Challenge-Response-Authentizierung – unterstützt. Das Kennwort für den Mail-Server wird also bei jeder Abfrage der Mailbox unverschlüsselt über das Netzwerk zum POP3-Server gesendet und kann von anderen Netzteilnehmern mit relativ geringem technischen Aufwand abgefangen werden. Abgesehen davon nimmt das Paßwort innerhalb der Konfigurationsdaten auch keine Sonderstellung ein und läßt sich von den dazu berechtigten Personen zusammen mit dem Rest der Konfiguration einfach aus dem Gerät abfragen.

Das eigentliche Sicherheitsrisiko liegt weniger in der Möglichkeit des unberechtigten Zugriffs auf das Postfach des IO-Mailers, denn Mails, die dort liegen, enthalten normalerweise keine großen Geheimnisse. Sie sollten vielmehr darauf achten, daß aus der Kenntnis des Paßworts möglichst wenige weitere Rechte erlangt werden können. Insbesondere sollte man hier nicht das Paßwort eines menschlichen Benutzers „wiederverwenden“.

Das Abfrageintervall schließlich bestimmt, wie schnell der IO-Mailer auf eingehende Mail-Kommandos reagieren kann. Um unnötige Netzwerkbelastung zu vermeiden, sollte man hier nicht zu ehrgeizig sein. Eine Reaktionszeit in der Größenordnung von einer Minute reicht im allgemeinen aus.

## 4.4 Absenderadresse des Mailers <device-id>

Beispiel:

```
<io-mail.1>
  <device-id>
    <host>ebox1.firma.de</host>
    <address>ebox1@firma.de</address>
    <name>E/A-Server im Treppenhaus</name>
    <owner>f.mustermann@firma.de</owner> <!-- optional -->
  </device-id>
</io-mail.1>
```

Diese Informationen verwendet der IO-Mailer beim Senden von Mail. Den eigenen Hostnamen gibt er an, um sich gegenüber dem SMTP-Server zu identifizieren. Die Mail-Adresse wird als Absender auf allen erzeugten Mails vermerkt und ermöglicht es dem Empfänger, dem IO-Mailer zu antworten (hilfreich vor allem für das Abbestellen unerwünschter Benachrichtigungen). Der „Name“ des Mailers schließlich ist ein beliebiger Text, der beim Empfänger anstelle von oder zusammen mit der Absenderadresse angezeigt wird. Hier könnten Sie einen kurzen Hinweis auf Funktion, Standort oder Betreiber des Mailers eingeben.

Das Feld <owner> sollte die Mail-Adresse einer Person enthalten, die für den Betrieb des IO-Mailers verantwortlich ist. Erstens wird diese Adresse in den Kopfzeilen aller automatisch erzeugten Mails vermerkt; wer bei etwaigen Problemen einen Ansprechpartner sucht, wird mit großer Wahrscheinlichkeit diesen Abschnitt einer Nachricht in Augenschein nehmen. Außerdem schickt der IO-Mailer an die hier angegebene Adresse Benachrichtigungen über besondere Vorkommnisse (z. B. daß ein Empfänger sich mit dem Mail-Kommando „unsubscribe“ selber aus der Konfiguration gelöscht hat). In vielen Fällen ist diese Angabe allerdings überflüssig – vor allem dann, wenn Sie den IO-Mailer nicht über die serielle Schnittstelle, sondern stets per Mail konfigurieren. In dem Fall wird als Ansprechpartner automatisch derjenige eingetragen, der zuletzt den Konfigurationszweig <alarm> geändert hat.

## 4.5 Erscheinungsbild des Gerätestatus <report>

Jede Mail, die aufgrund eines Ereignisses (oder als Antwort auf das Kommando „status“) versandt wird, enthält als Text mindestens eine Auflistung der sechs Gerätevariablen. Im einfachsten Fall kann sie also etwa folgendermaßen aussehen:

```
Thu, 29 Jun 2000 13:52:44
X0 = 0
X1 = 1
```

```
Y0 = 0
Y1 = 0
N0 = 37
N1 = 4
```

Weil dies für den Empfänger der Mail ziemlich nichtssagend ist, kann jeder Variable ein Kommentartext zugeordnet werden; den Binärvariablen X0, X1, Y0, Y1 außerdem je ein Text für den Zustand 0 und 1.

Abgesehen davon läßt sich über diesen Abschnitt der Konfiguration auch noch die Aufnahme von Meßreihen steuern: Für die beiden Zähler N0 und N1 können bis zu 256 zurückliegende Werte in einem frei definierbaren Zeitraster gespeichert werden.

Hier ein komplettes Beispiel:

```
<io-mail.1>
  <report>
    <label var="X0" alias="Lichtschranke"
      state0="frei" statel="unterbrochen"/>
    <label var="X1" alias="Klingeltaster"
      state0="losgelassen" statel="betätigt"/>
    <label var="Y0" alias="Beleuchtung" state0="aus" statel="ein"/>
    <label var="N0" alias="Besucher"/>
    <history var="N0" depth="4" step="00:00:15"/>
    <history var="N1" depth="12" step="00:00:05"/>
  </report>
</io-mail.1>
```

Und eine mögliche Ausgabe, die dadurch erzeugt würde:

```
Wed, 2 Aug 2000 17:02:04
X0 = 0 (Lichtschranke frei)
X1 = 0 (Klingeltaster losgelassen)
Y0 = 1 (Beleuchtung ein)
Y1 = 0
N0 = 6 (Besucher)
N1 = 10

17:01:49, N0 = 5 (+1)
17:01:34, N0 = 2 (+3)
17:01:19, N0 = 0 (+2)
17:01:04, N0 = 0

17:02:00, N1 = 9 (+1)
17:01:55, N1 = 9
17:01:50, N1 = 6 (+3)
17:01:45, N1 = 6
17:01:40, N1 = 6
17:01:35, N1 = 4 (+2)
17:01:30, N1 = 0 (+4)
```



```
17:01:25, N1 = 0
17:01:20, N1 = 0
17:01:15, N1 = 0
17:01:10, N1 = 0
17:01:05, N1 = 0
```

Die Zahlen in Klammern hinter den einzelnen Meßreihen-Werten geben dabei jeweils die Differenz zum nächstjüngeren Wert an. Im Falle der ersten Zeile jeder Liste ist das die Differenz zu dem aktuellen Wert, der nicht mehr in der Liste selber, sondern in der Zusammenfassung der Variablen am Anfang der Mail steht.

## 4.6 Zeitzone <zone>

Gibt die Zeitzone an, in der sich der IO-Mailer befindet, z.B. +0100 für MEZ (Mittel-europäische Zeit), +0200 für MESZ (MEZ Sommerzeit). Wie durch diese Beispiele bereits angedeutet, wird hier jeweils zweistellig und ohne Trennzeichen die Stunden-und-Mi-nuten-Verschiebung gegenüber der Weltzeit angegeben (GMT, früher auch als Green-wich-Zeit bezeichnet). Beispiel:

```
<io-mail.1>
<zone>+0100</zone>
</io-mail.1>
```

Wichtig: Die Umschaltung zwischen Sommer- und Winterzeit erfolgt nicht automatisch, sondern muß vom Benutzer selbst eingegeben werden.

Der IO-Mailer verwendet die Zonen-Information, um die Weltzeit, wie er sie von seinem Timeserver erhält, in Ortszeit umzurechnen. Durch die Angabe einer falschen Zeitzone ist die Uhrzeit im Statusbericht des Mailers gegenüber der tatsächlichen Ortszeit entsprechend verschoben.

Sofern die vom Timeserver empfangene Weltzeit korrekt ist, läßt sich aber auch bei falscher Zoneninformation noch feststellen, wann eine Mail wirklich gesendet wurde. In den Kopfzeilen der Mail ist nämlich nicht nur die vermeintliche Ortszeit selber vermerkt, sondern auch die Zeitzone, für die diese Ortszeit gelten soll. Einige Mail-Clients werten diese Informationen bereits aus, um den Zeitpunkt, zu dem eine Mail gesendet wurde, sofort auf die Ortszeit des Empfängers umzurechnen.

## 4.7 Zugriffsbeschränkungen <access>

Bevor der IO-Mailer ein Mail-Kommando ausführt, prüft er, ob der Absender der Mail zu der betreffenden Aktion berechtigt ist. Dazu lassen sich zwei bevorrechtigte Perso-nengruppen einrichten: Benutzer (*user*) und Administratoren (*admin*). Vereinfacht ge-sagt braucht es mindestens Benutzerrechte, um irgend etwas am IO-Mailer zu verändern, und es braucht Administratorrechte, um kritische Details der Konfiguration zu ändern. (Als „kritisch“ gelten dabei solche Änderungen, mit denen man sich selber aussperren kann, die sich also möglicherweise nicht mehr per Mail, sondern nur noch vor Ort am

seriellen Terminal rückgängig machen lassen.) Eine dritte bevorrechtigte Personengruppe sind die vertrauenswürdigen Empfänger, zur der aber von vornherein jeder gehört, der nicht ausdrücklich als *suspect* ausgeschlossen wurde. Im Detail sieht das so aus:

jedermann	help status
vertrauenswürdiger Empfänger	unsubscribe resume
Benutzer	set info template config ( <i>mit den Zweigen</i> report, alarm, sleep, device_id, zone)
Administrator	update config ( <i>mit den Zweigen</i> access, servers, pop_account)

Eine sinnvolle Politik könnte z.B. darin bestehen, allen Mitarbeitern der eigenen Firma Benutzerrechte und nur einigen ausdrücklich genannten Firmenangehörigen darüber hinaus auch Administratorrechte einzuräumen. So ist es in der folgenden Konfiguration festgelegt:

```
<io-mail.1>
<access>
<user>firma.de</user>
<admin>f.mustermann@firma.de</admin>
<admin>k.zuse@firma.de</admin>
</access>
</io-mail.1>
```

Einträge, die ein „@“ enthalten, stellen dabei einfache Mail-Adressen dar; Einträge ohne „@“ gelten für alle Absender aus einer kompletten Mail-Domain. Durch den Eintrag „firma.de“ würden hier also z.B. „frank@firma.de“ und „klaus@firma.de“ Benutzerrechte eingeräumt, ebenso „thomas@zweigstelle.firma.de“, nicht aber „stefan@andere-firma.de“ oder „martin@firma.definitiv.de“.

Eine Schwachstelle bei all diesen schönen Überlegungen bleibt jedoch, daß sich Mail-Absenderadressen fälschen lassen. Wer so etwas tut, muß aber einige Probleme überwinden, z.B. werden ihn die Kommando-Quittungen des IO-Mailers nicht mehr erreichen. (Quittungen werden stets an die Adresse geschickt, die auch zur Prüfung der Berechtigung verwendet wurde; *Reply-To*-Kopfzeilen werden also nicht beachtet.) An der grundsätzlichen Möglichkeit eines solchen Angriffs ändert das aber nichts.

Wenn Sie möchten, konfigurieren Sie den IO-Mailer deshalb so, daß sich Änderungen nur noch direkt über seinen seriellen Port vornehmen lassen:

```
<io-mail.1>
<!-- keine bevorrechtigten Zugriffe per Mail mehr erlauben -->
<access>
<suspect>*</suspect>
</access>
</io-mail.1>
```

Das „\*“ steht dabei als Joker, der auf alle Adressen paßt und in diesem Fall alle Mail-Teilnehmer in die niedrigste Berechtigungsstufe einordnet. Eine anschauliche Deutung der Benutzergruppe *suspect* wäre etwa „Empfänger, denen nicht einmal das *unsubscribe*-Kommando erlaubt werden kann, weil sie es doch nur mißbrauchen würden“, was sicher etwas übertrieben klingt. Aber es geht in diesem Beispiel darum, Sabotage durch Mails mit gefälschter Absenderadresse zu verhindern. Das Mißtrauen richtet sich nicht wirklich gegen die Personen, die als Empfänger eingetragen sind.

## 4.8 Zeitabhängige Steuerung <schedule>

Die Aktionen des IO-Mailers lassen sich in begrenztem Umfang auch in Abhängigkeit von Tageszeit und Wochentag steuern. Dazu dienen zwei zusätzliche binäre Variablen *WE* und *NT*, die den Zustand „Wochenende“ bzw. „Nacht“ anzeigen. Um sie zu nutzen, müssen Sie zunächst festlegen, welche Zeiträume als Nacht und Wochenende gelten sollen:

```
<io-mail.1>
  <schedule>
    <night from="22:00:00" to="06:00:00"/>
    <weekend from="Fr 22:00:00" to="Mo 06:00:00"/>
  </schedule>
</io-mail.1>
```

Anmerkungen:

Die Angabe des Wochentages kann mit deutschen oder englischen Abkürzungen erfolgen, wobei lediglich die ersten zwei Buchstaben beachtet werden. So wären z.B. „*Di*“, „*Die*“, „*Tu*“ und „*Tue*“ allesamt gültige Bezeichnungen für Dienstag.

Solange die Echtzeituhr des IO-Mailers nicht gestellt werden konnte (weil z.B. keiner der konfigurierten Timeserver erreichbar war oder weil überhaupt keiner angegeben wurde), ist die zeitabhängige Steuerung unwirksam, und die Variablen *WE* und *NT* haben den Wert 0.

Die zeitabhängigen Variablen werden auch im Gerätestatus angezeigt – allerdings nur wenn sie überhaupt benutzt werden und (wie hier gerade gezeigt) Zeiten für ihre Funktion festgelegt sind. Für das Erscheinungsbild des Gerätestatus würden sich dann zwei zusätzliche Zeilen im <report>-Zweig anbieten, die ungefähr so aussehen könnten:

```
<label var="WE" state0="Werktag" state1="Wochenende"/>
<label var="NT" state1="Nacht"/>
```

## 4.9 Ereignisse und Aktionen <alarm>

Über diesen Teil der Konfiguration wird die eigentliche Nutzenanwendung des IO-Mailers gesteuert. Das Grundprinzip dabei ist, daß Änderungen an den Eingängen ihrerseits Änderungen an den Ausgängen und/oder das Senden einer Mail bewirken können.

Genauer betrachtet haben wir es mit Variablen, Operatoren, Triggerbedingungen, Ereignissen, Aktionen und Empfängerlisten zu tun. Diese Begriffe sollen im folgenden kurz erläutert werden.

### Variablen

Es gibt vier binäre Variablen, die den Eingängen ( $X0$ ,  $X1$ ) und Ausgängen ( $Y0$ ,  $Y1$ ) zugeordnet sind und jeweils die Zustände 0 und 1 annehmen können. Außerdem überwacht der IO-Mailer in zwei Zählervariablen ( $N0$ ,  $N1$ ) die Änderungen an den Eingängen. Gezählt werden dabei nicht Flanken, sondern Zyklen. Anders ausgedrückt: Betätigen und Loslassen eines Tasters an einem Eingang erhöht den zugehörigen Zähler nicht um zwei, sondern um eins. Die genaue Regel lautet, daß jedesmal gezählt wird, wenn ein Eingang zu dem Zustand zurückkehrt, den er beim Einschalten des Gerätes hatte. Im obigen Beispiel würde sich der Zählerstand also erst beim Loslassen des Tasters ändern, und zwar unabhängig davon, ob dieser Taster einen Kontakt öffnet oder schließt. Ob nun wirklich bei steigender oder bei fallender Flanke gezählt wird, dürfte in den meisten Anwendungsfällen aber ohnehin keinen großen Unterschied machen.

Zwei weitere Binärvariablen zur Anzeige von Nacht und Wochenende ( $NT$ ,  $WE$ ) wurden bereits im vorigen Abschnitt erwähnt.

### Operatoren

Es gibt sechs einfache Operatoren, die jeweils eine Variable gegen einen konstanten Wert vergleichen:

EQ	Gleichheit	"equal"
NE	Ungleichheit	"not equal"
GT	größer als	"greater than"
LT	kleiner als	"less than"
GE	größer oder gleich	"greater or equal"
LE	kleiner oder gleich	"less or equal"

Zusätzlich gibt es noch den siebten „*DELTA*“-Operator. Seine Testbedingung sind erfüllt, sobald sich der zugehörige Variablenwert um einen Mindestbetrag geändert hat. (Zu diesem Zweck merkt sich der Operator einen Momentanwert seiner Variablen als Referenz. Wann genau er diese Referenzwerte aufnimmt, wird weiter unten bei der Erläuterung der Ereignisse erklärt.)

### Triggerbedingungen

Eine Triggerbedingung faßt im wesentlichen einen Operator mit seinen zugehörigen Vergleichsobjekten „Variable“ und „konstante Zahl“ zusammen.

Darüber hinaus kann bei Bedarf noch eine Ansprechverzögerung (*delay*) angegeben werden. Sie bewirkt, daß das Testergebnis des Operators erst über eine vorgegebene Zeitspanne hin ununterbrochen erfüllt sein muß, bevor die Triggerbedingung selbst erfüllt ist. Ebenso läßt sich noch eine Erholzeit (*recover*) angeben, die umgekehrt bewirkt, daß eine Triggerbedingung eine vorgegebene Zeitspanne hin weiter erfüllt bleibt, auch wenn das Testergebnis des Operators nicht mehr erfüllt ist. Diese Parameter sollen verhindern, daß ein geschlossenes Ereignis infolge vorübergehender Störungen (etwa

durch prellende Kontakte) als Abfolge mehrerer Ereignisse registriert wird. Allerdings werden in der Praxis viele Störsignale auch bereits dadurch ausgefiltert, daß die Eingangsvariablen des IO-Mailers nur in einem Zeitraster von 10 ms aktualisiert werden.

Ein Beispiel: Angenommen, an Eingang *X0* ist ein Taster angeschlossen. Dann ist die erste der folgenden Triggerbedingungen erfüllt, nachdem der Taster zehnmals betätigt wurde, die zweite, wenn man ihn mindestens eine halbe Sekunde lang gedrückt hält:

```
<trigger op="DELTA" var="N0" val="10"/>  
<trigger op="EQ" var="X0" val="1" delay="0.5"/>
```

### Aktionen

Eine „Aktion“ im Sinne des IO-Mailers ist eine Wertzuweisung an eine seiner Variablen. (Das Versenden einer Mail fällt also nicht unter diesen Begriff.) Am häufigsten dürfte dabei das Setzen von Ausgangsvariablen (*Y0*, *Y1*) sein, doch können auch die Zählerstände (*N0*, *N1*) auf Wunsch geändert werden. Nicht erlaubt sind dagegen Zuweisungen an die Tageszeitvariablen und die Eingänge.

### Empfängerlisten

Empfängeradressen können einem bestimmten Ereignis zugeordnet sein oder auch global gelten. Eine Mail wird versandt, wenn ein Ereignis eintritt, zu dem mindestens eine Empfängeradresse angegeben ist. Falls es global gültige Adressen gibt, löst dementsprechend jedes Ereignis eine Mail aus.

### Ereignisse

Ein Ereignis besteht aus Triggerbedingungen, Aktionen und einer Liste von Mail-Empfängern. Bei mehreren Triggerbedingungen müssen alle gleichzeitig erfüllt sein, um das Ereignis auszulösen (implizite UND-Verknüpfung).

Sobald ein Ereignis eintritt, werden die zugehörigen *Aktionen* unverzüglich ausgeführt (mit einer Reaktionszeit in der Größenordnung von 10 ms), evtl. an den Triggerbedingungen beteiligte *DELTA*-Operatoren werden zurückgesetzt (indem sie angewiesen werden, einen neuen Referenzwert aufzunehmen), und es wird eine Mail-Anforderung vermerkt, die erst wieder gelöscht wird, wenn tatsächlich eine entsprechende Mail erzeugt werden konnte. Auch wenn der Mail-Versand sich aus verschiedenen Gründen erheblich verzögern kann, ist also sichergestellt, daß keine Mails vergessen werden, nur weil etwa das zugehörige Ereignis (z.B. Betätigung eines Tasters) zu schnell wieder beendet wäre. Mail-Anforderungen sammeln sich aber nicht an: Wenn z.B. ein Ereignis mehrmals eintritt, bevor sich Gelegenheit zum Senden bietet, wird trotzdem nur eine einzige Mail erzeugt.

All dies geschieht wie in der Regel nur, wenn ein Ereignis gerade neu eintritt, läßt sich aber auch mit Hilfe eines zusätzlichen *repeat*-Parameters in regelmäßigen Abständen wiederholen, solange das Ereignis andauert (d. h. solange alle seine Triggerbedingungen erfüllt sind). Beachten Sie dabei nur, daß Ereignisse, die einen *DELTA*-Operator enthalten, naturgemäß niemals andauern können und sich darum auch nicht wiederholen lassen.

## 4.9.1 Beispiele

In diesem ersten Beispiel sind zwei Ereignisse definiert, die eintreten, wenn ein Signal an Eingang *X0* bzw. *X1* angelegt wird. Die Mails, die erzeugt werden, unterscheiden sich durch ihre Betreffzeilen, aber Empfänger und Meldungstext sind in beiden Fällen gleich. Eine Besonderheit stellt noch der Meldungstext „,“ dar. Der gesamte Text hinter einem solchen einzelnen Punkt wird beim Versand der Mail abgeschnitten – also auch der Statusbericht, der normalerweise an jede Mail angehängt wird (vgl. Beschreibung zum Konfigurationszweig *report*). Um den Empfänger nicht mit unnötigen Details zu verwirren, kann dies in manchen Fällen wünschenswert sein.

```
<io-mail.1>
  <alarm>
    <event>
      <trigger op="EQ" var="X0" val="1"/>
      <subject>Bitte Kaugummiautomaten nachfüllen</subject>
    </event>
    <event>
      <trigger op="EQ" var="X1" val="1"/>
      <subject>Bitte Zigarettensautomaten nachfüllen</subject>
    </event>
    <!-- die folgenden Angaben gelten für beide event's -->
    <mailto>f.mustermann@firma.de</mailto>
    <message>Wegen Schlüssel beim Hausmeister melden.</message>
    <message>.</message>
  </alarm>
</io-mail.1>
```

Wie bereits erwähnt lassen sich auch mehrere Triggerbedingungen miteinander verknüpfen. Das ist vor allem sinnvoll im Zusammenhang mit den zeitabhängigen Variablen (*WE*, *NT*), deren Änderungen für sich alleine sicher nicht besonders interessant sind:

```
<io-mail.1>
  <alarm>
    <event>
      <!-- Die beiden folgenden Bedingungen müssen gleichzeitig -->
      <!-- erfüllt sein (implizite UND-Verknüpfung): -->
      <trigger op="EQ" var="X0" val="1"/>
      <trigger op="EQ" var="WE" val="1"/>
      <subject>Störung in der Heizungsanlage</subject>
      <message>Und das ausgerechnet am Wochenende...</message>
    </event>
    <mailto>f.mustermann@firma.de</mailto>
  </alarm>
</io-mail.1>
```

Das nächste Beispiel führt etwas kompliziertere Triggerbedingungen vor. Stellen Sie sich vor, daß Ausgang *Y0* ein Lämpchen ansteuert. Die ersten beiden Ereignisse arbeiten zusammen und bewirken, daß dieses Lämpchen blinkt (0.3 Sek. ein, 0.2 Sek. aus), solange ein Signal an Eingang *X0* anliegt. Das dritte dazugehörige Ereignis stellt sicher, daß das Lämpchen erlischt, wenn das Eingangssignal an *X0* endet. Das vierte Ereignis schließlich würde auch unabhängig von den beiden anderen funktionieren. Es verschickt Benachrichtigungen, solange das Signal an *X0* anliegt – erstmals nach einer Sekunde, danach im Abstand von 10 Minuten.

```
<io-mail.1>
  <alarm>
    <event>
      <trigger op="EQ" var="X0" val="1"/>
      <trigger op="EQ" var="Y0" val="0" delay="0.2"/>
      <action var="Y0" val="1"/>
    </event>
    <event>
      <trigger op="EQ" var="X0" val="1"/>
      <trigger op="EQ" var="Y0" val="1" delay="0.3"/>
      <action var="Y0" val="0"/>
    </event>
    <event>
      <trigger op="EQ" var="X0" val="0"/>
      <action var="Y0" val="0"/>
    </event>
    <event>
      <trigger op="EQ" var="X0" val="1" delay="1"/>
      <repeat>00:10:00</repeat>
      <mailto>f.mustermann@firma.de</mailto>
      <subject>bitte Behälter ausleeren</subject>
      <message>.</message>
    </event>
  </alarm>
</io-mail.1>
```

Man kann den IO-Mailer auch dazu bringen, eine Einschaltmeldung zu verschicken, z. B. um zu überwachen, ob und wann vorübergehende Ausfälle der Versorgungsspannung auftreten. Zu dem Zweck formuliert man ein Ereignis ohne jede Triggerbedingung. Ein solches Ereignis tritt dann nur einmal ein (weil es keine Triggerbedingung gibt, die nicht erfüllt wäre) und danach nie wieder (weil es eben *andauert*, was erneutes Eintreten ausschließt).

```
<io-mail.1>
  <alarm>
    <event>
      <mailto>f.mustermann@firma.de</mailto>
      <subject>E/A-Server startet</subject>
    </event>
```

```
...
</alarm>
</io-mail.1>
```

Eine leichte Abwandlung dieser Technik läßt sich auch verwenden, um in regelmäßigen Abständen Statusberichte zu verschicken. Der *repeat*-Parameter bewirkt in diesem Beispiel, daß das Ereignis, solange seine Triggerbedingungen erfüllt sind (und das sind sie ihrer Natur nach immer), in Abständen von einer Stunde immer wieder erneut eintritt.

```
<io-mail.1>
<alarm>
  <event>
    <mailto>f.mustermann@firma.de</mailto>
    <subject>Keine besonderen Vorkommnisse</subject>
    <message>Nur schon wieder eine Stunde um.</message>
    <repeat>01:00:00</repeat>
  </event>
</alarm>
</io-mail.1>
```

## 4.9.2 Einschränkungen

- Als Triggerbedingung können nicht zwei Variablen miteinander verglichen werden, sondern stets nur eine Variable mit einem konstanten Wert.
- In der vorliegenden Firmware-Version können maximal 10 Ereignisse, 20 Triggerbedingungen und 40 Aktionen definiert werden.
- Die größtmöglichen Werte für Ansprechverzögerung und Erholzeit einer Triggerbedingung sind 00:10:55, also knapp elf Minuten.
- Meldungstexte können höchstens 254 Zeichen lang sein (das entspricht ca. drei bis vier Zeilen). Der Inhalt einer Mail kann zwar aus mehreren *message*-Blöcken bestehen, diese Texte werden aber nicht nahtlos aneinandergehängt, sondern als eigene Absätze dargestellt. Außerdem steht für alle Konfigurationsdaten zusammen nur ein Speicherbereich von 8 KByte zur Verfügung.

## 4.10 Diverse mögliche Probleme

- Da es oftmals möglich ist, einen Mail-Empfänger unter mehreren Varianten ein- und derselben Adresse zu erreichen (z.B. „f.mustermann@mail.firma.de“ als Äquivalent zu „f.mustermann@firma.de“), kann es sein, daß nicht alle Empfänger das „unsubscribe“-Kommando verwenden können. Es wird für solche Empfänger fehlgeschlagen, die in der Konfiguration unter einer anderen Adresse als derjenigen eingetragen sind, die ihre eigene Mail-Software als Absenderadresse vermerkt.
- Das XML-Dokument mit den Konfigurationsdaten darf höchstens 20 KByte lang sein; dabei spielt es keine Rolle ob es per Mail oder lokal über den seriellen Port eingespielt wird. Da Kommentare hier aber nicht mitgerechnet werden, müssen Sie sich in dieser Hinsicht nicht unbedingt einschränken. Allerdings gibt es darüber hinaus für Mails eine Beschränkung auf 64 KByte Gesamtgröße; noch umfangrei-



chere Mails werden ungelesen gelöscht. Dieser Grenzwert ist grundsätzlich notwendig, da der IO-Mailer sonst durch (böswillig oder irrtümlich gesendete) riesige Dateianhänge über lange Zeiträume blockiert werden könnte.

- Anders als in der XML-Spezifikation verlangt versteht der IO-Mailer nicht die 16-bit-Zeichensätze UTF-8 und UTF-16, sondern nur den 8-bit-Zeichensatz ISO-8859-1 sowie dessen Teilmenge 7-bit-US-ASCII. Da ISO-8859-1 der in Westeuropa übliche Zeichensatz unter Windows und Unix ist, wird diese Einschränkung normalerweise keine Probleme bereiten, solange Sie einen einfachen Texteditor zum Bearbeiten der Konfigurationsdaten verwenden. Wenn Sie aber einen speziellen XML-Editor benutzen, sollten sie darauf achten, daß Sie die Daten keinesfalls als UTF-16 speichern. Nach Möglichkeit sollten Sie hier ISO-8859-1 verwenden und nur zur Not auf UTF-8 zurückgreifen.
- Der IO-Mailer versteht nicht das MIME-Codierverfahren base64, sondern nur quoted-printable und darüber hinaus natürlich einfachen Klartext. Vernünftige Mailsoftware sollte keinen Anlaß sehen, die Konfigurationsdaten jemals als base64 zu codieren. Falls dies aber dennoch geschieht, werden die Konfigurationsdaten mit der Fehlermeldung „*start tag <io-mail.1> not found*“ abgelehnt. Versuchen Sie in diesem Fall, das Verhalten Ihrer Mailsoftware zu beeinflussen, indem Sie die Konfigurationsdaten nicht als Dateianhang schicken, sondern direkt in den Textkörper der Mail einfügen.

## 5. Anhang: Internetadressen

### 5.1 Zum Thema XML

Allgemeines über XML (Spezifikation, Fragen und Antworten):

<http://www.w3.org/TR/REC-xml>

<http://msdn.microsoft.com/xml/tutorial/default.asp>

<http://www.ucc.ie/xml/>

<http://msdn.microsoft.com/workshop/xml/general/xmlfaq.asp>

Gratis-Download: Der Microsoft XML-Validator (benötigt Internet-Explorer 5). Sie erhalten im wesentlichen eine HTML-Seite, in die Auszüge von JavaScript integriert sind. Damit ist es möglich, den Namen einer XML-Datei einzulesen und diese zur Überprüfung an die eingebauten Routinen des Explorers weiterzuleiten. Sehr hilfreich, wenn man XML-Dateien bereits vor Versenden an den IO-Mailer auf korrektes Format prüfen will.

[http://msdn.microsoft.com/downloads/samples/internet/xml/xml\\_validator/default.asp](http://msdn.microsoft.com/downloads/samples/internet/xml/xml_validator/default.asp)

Gratis-Download: Das Microsoft XML-Notepad (benötigt ebenfalls Internet-Explorer 5). Hübsches Spielzeug, ist zum Bearbeiten von Konfigurationsdateien für den IO-Mailer aber keinesfalls erforderlich. Das geht genauso gut mit einem gewöhnlichen Texteditor.

<http://msdn.microsoft.com/xml/notepad/intro.asp>

### 5.2 Zum Thema Timeserver

Ein Verzeichnis weltweit erreichbarer Timeserver (eigentlich: NTP-Server) finden Sie unter <http://www.eecis.udel.edu/~mills/ntp/servers.htm>.

NTP (*Network Time Protocol*) ist ein aufwendiges Protokoll zur möglichst präzisen Uhrensynchronisation in weitläufigen Netzwerken (s. RFC-1305), nicht zu verwechseln mit dem einfachen Zeitanagedienst nach RFC-868, den der IO-Mailer benötigt. In der Praxis bieten die meisten NTP-Server aber auch den einfachen Zeitanagedienst zusätzlich an. Beachten Sie bitte, daß Sie nur solche Server aus dieser Liste verwenden sollten, die ausdrücklich als „Open access“ gekennzeichnet sind.

Gratis-Download: Timeserver für Windows-Rechner. Für die meisten Zwecke dürfte eine solche lokale Lösung sinnvoller sein, als auf einen fremden Timeserver im Internet zuzugreifen.

<http://www.arachnoid.com/abouttime/index.html>

Wie bereits weiter oben erwähnt, können Sie auf derartige Software verzichten, wenn Sie einen eigenen Unix- oder Linux-Server besitzen. Dort ist ein Timeserver nach RFC-868 bereits eingebaut; Sie sollten lediglich sicherstellen, daß er auch aktiviert ist (Dienst namens „time“ in der Datei „inetd.conf“).

## Technische Daten Typen 57111, 57211

Spannungsversorgung:	5V über Steckernetzteil 230V/50Hz	
Stromaufnahme:	Typ 57111	57211
typisch	270mA	455mA
max.	330mA	575mA
Konfigurationsschnittstelle:	RS232 DCE-Belegung auf DB9F 9600Bd, 8 Bit, no Parity (nur RxD und TxD unterstützt)	
Digitale Eingänge:	2 Eingänge, spannungsfest bis +/-30V CMOS-kompatible Schaltschwelle mit 32-Bit-Zählfunktion (max. +/-2147483647)	
Digitale Ausgänge:	2 Ausgänge „Open Collector“ max. 30V/100mA	
Netzwerk:	Ethernet 10BaseT Ethernet 100BaseTx/10BaseT	
Mailfunktionen:	Konfiguration, Alarmierung, Berichtsfunktionen und Setzen der Ausgänge	
Mailprotokolle:	POP3, SMTP	
Gehäuse:	Kunststoffkleingehäuse, 105x75x22mm geeignet für Hutschienenmontage	
Gewicht:	ca. 200g	
Umgebungstemperatur Betrieb:	0..+50°C bei angereichten Gehäusen 0..+60°C bei Einzelbetrieb	
Umgebungstemperatur Lagerung:	-40..+70°C	

Irrtum und Änderung vorbehalten

Wiesemann & Theis GmbH

## Technische Daten Typen 57511, 57611

Spannungsversorgung:	12–24V AC oder DC über Schraubklemme
Stromaufnahme (ohne Beschaltung der Ein- und Ausgänge):	Typ 57511      Typ 57611
typisch	155mA          175mA @24V
max.	205mA          225mA @24V
Konfigurationsschnittstelle:	RS232 DCE-Belegung auf DB9F 9600Bd, 8 Bit, no Parity (nur RxD und TxD unterstützt)
Digitale Eingänge:	2 Eingänge, spannungsfest bis +/-30V CMOS-kompatible Schaltschwelle mit 32-Bit-Zählfunktion (max. +/-2147483647)
Digitale Ausgänge:	2 Ausgänge „Open Collector“ max. 30V/100mA
Netzwerk:	Typ 57511 Ethernet 10BaseT Typ 57611 Ethernet 100BaseTx/10BaseT
Mailfunktionen:	Konfiguration, Alarmierung, Berichtsfunktionen und Setzen der Ausgänge
Mailprotokolle:	POP3, SMTP
Gehäuse:	Kunststoffkleingehäuse, 105x75x22mm geeignet für Hutschienenmontage
Gewicht:	ca. 200g
Umgebungstemperatur Betrieb:	0..+50°C bei angereichten Gehäusen 0..+60°C bei Einzelbetrieb
Umgebungstemperatur Lagerung:	-40..+70°C

Irrtum und Änderung vorbehalten

Wiesemann & Theis GmbH