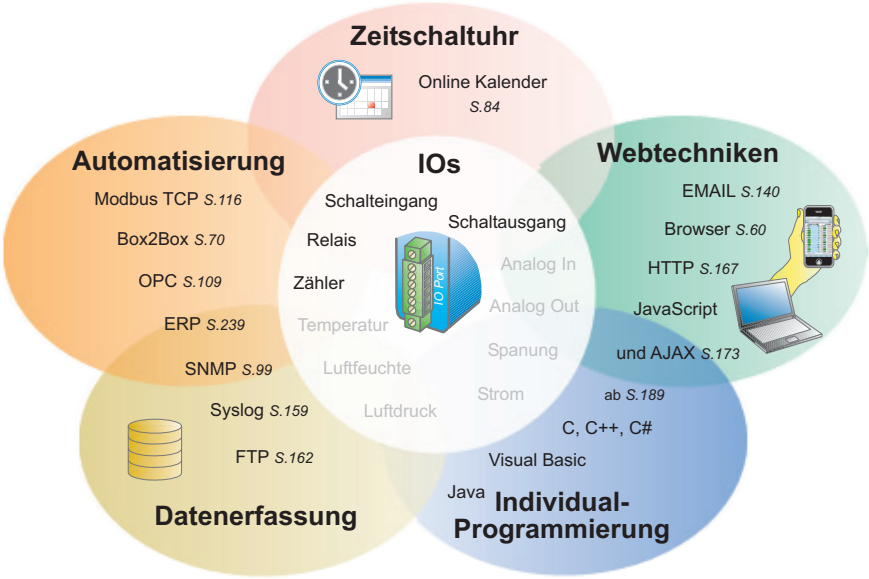
# Referenzhandbuch




**gültig für folgende Web-IO Modelle:**

**#57637** Web-IO 2xDigital In, 2xDigital Out
**#57630M** Web-IO 12xDigital In, 12xDigital Out
**#57631M** Web-IO 12xDigital In, 12xDigital Out mit integriertem Com-Server
**#57634M** Web-IO 12xDigital In, 6xRelaisOut
**#57638** ERP-Gateway 2xDig. In, 2xDig. Out

**Release** 3.45, April. 2014

ab Firmware 3.45

W&T

© 01/2011 by Wiesemann und Theis GmbH
Microsoft, MS-DOS, Windows, Winsock und Visual Basic sind eingetragene Warenzeichen der Microsoft Corporation

Irrtum und Änderung vorbehalten:
Da wir Fehler machen können, darf keine unserer Aussagen ungeprüft verwendet werden. Bitte melden Sie uns alle Ihnen bekannt gewordenen Irrtümer oder Missverständlichkeiten, damit wir diese so schnell wie möglich erkennen und beseitigen können.

Führen Sie Arbeiten an bzw. mit W&T Produkten nur aus, wenn sie hier beschrieben sind und Sie die Anleitung vollständig gelesen und verstanden haben. Eigenmächtiges Handeln kann Gefahren verursachen. Wir haften nicht für die Folgen eigenmächtigen Handelns. Fragen Sie im Zweifel lieber noch einmal bei uns bzw. bei Ihrem Händler nach!

## Inhalt


**1. Übersicht der Einsatzmöglichkeiten 9**
1.1 Übersicht der Einsatzmöglichkeiten ..... 10
1.1.1 Browser-Zugriff ..... 10
für PC-Browser ..... 10
für das Smartphone ..... 10
Eigene Webseiten ..... 10
1.1.2 Box-to-Box ..... 10
1.1.3 Zeitgesteuertes Schalten ..... 10
Nutzung eines lokalen Kalenderprogramms ..... 11
Schaltzeiten im Google Kalender planen ..... 11
1.1.4 Integration in bestehende Systeme ..... 11
OPC ..... 11
Modbus TCP ..... 11
SNMP - Simple Network Management Protocol ..... 11
1.1.5 Zugriff aus eigenen Anwendungen ..... 12
Kommunikation per HTTP-Requests ..... 12
Kommunikation per Kommandostring ..... 12
Kommunikation über Binärstrukturen ..... 12
1.1.6 Email und mehr - Info bei Zustandsänderung ..... 12
1.1.7 Erfassen von Schaltzuständen ..... 13
Meldungen über das Web-IO Alarmsystem ..... 13
IO-Zustände in W&T-Motherbox erfassen ..... 13
1.1.8 Logik Funktionen - In- und Outputs verknüpfen ..... 13
1.1.9 Basisverhalten der Ports ..... 14
Inputs ..... 14
Outputs ..... 14
1.1.10 Allgemeine Basisfunktion ..... 14

**2 Vorbereitungen zur Inbetriebnahme 15**
2.1 Versorgungsspannung und Netzwerkanschluss ..... 16
2.1.1 Web-IO #57637 und #57638 ..... 16
2.1.2 Web-IO #57630, #57631, #57634 und #57636 ..... 17

2.2 Vergabe der IP-Adresse ..... 18
2.2.1 IP-Adresse automatisch per DHCP beziehen ..... 18
2.2.2 Vergabe der IP-Adresse mit dem Wutility-Tool ..... 18

**3 Anschlüsse, Bedienelemente und LEDs 21**

3.1 Anschlüsse Web-IO 2x Dig. Input, 2x Dig. Output 22
3.1.1 Externe Spannungsversorgung ..... 22
3.1.2 Spannungsversorgung über PoE ..... 23
3.1.3 Input-Beschaltung ..... 23
3.1.4 Output-Beschaltung ..... 26

3.2 Anschlüsse Web-IO 12x Dig. In, 12xDig. Out ..... 29
3.2.1 Versorgungsspannungseingang ..... 29
3.2.2 Input-Beschaltung ..... 30
3.2.3 Output-Beschaltung ..... 31

3.3 Anschlüsse Web-IO 12x Dig. Input, 6x Relais ..... 34
3.3.1 Versorgungsspannungseingang ..... 34
3.3.2 Input-Beschaltung ..... 34
3.3.3 Output-Beschaltung ..... 37

3.4 Ethernet-Anschluss ..... 39

3.5 Serieller Anschluss ..... 41

3.6 LEDs für Gerätestatus ..... 42

**4 Konfiguration der Netzwerkparameter 45**

4.1 Web-Based Management ..... 46
4.1.1 Aufbau der Webseite ..... 46
4.1.2 Die Navigation ..... 46
4.1.3 Erleichterte Navigation durch Profile ..... 47
4.1.4 Zugriffsrechte ..... 47
4.1.5 Ändern und Einrichten der Zugriffspasswörter ..... 48
4.1.6 Übernahme und Aktivierung der Einstellungen ..... 49

4.2 Grundeinstellung der Netzwerkparameter ..... 51
Ändern des Device Name ..... 55

**5 Basisfunktionen 59**

5.1 HTTP - Inputs und Outputs im Browser steuern ... 60
5.1.1 Die Home-Seite ..... 60
5.1.2 Die User-Seite ..... 61
5.1.3 Die Smart-Seite ..... 63
5.1.4 Beschriftungen und Texte ..... 64
5.1.5 Ändern des HTTP-Ports ..... 68

5.2 Box-to-Box ..... 70

5.2.1 Konfiguration des Slave Web-IO ..... 70
5.2.2 Konfiguration des Masters ..... 74
5.2.3 Box-to-Box Verbindungsstatus ermitteln ..... 78
5.2.4 Beenden des Box-to-Box Modus ..... 78
5.2.5 Box-to-Box Slave - Bindung aufheben ..... 80
5.2.6 Box-to-Box Hinweise und Grenzen ..... 81

5.3 Time Switch - Web-IO als Schaltuhr ..... 84
5.3.1 Aktivieren er Outputs für Time Switch ..... 84
5.3.2 Schaltvorgaben laden ..... 85
5.3.3 Schaltzeiten aus Online-Kalender laden ..... 85
5.3.4 Schaltzeiten als iCal-Datei laden ..... 91
5.3.5 Schaltuhr Vorschau ..... 97
5.3.6 Besonderheiten der Web-IO Schaltuhr ..... 97

5.4 Integration in bestehende Systeme ..... 99
5.4.1 SNMP - In- und Outputs in Managementsystemen 99
Freischalten des SNMP-Zugriffs ..... 99
Aktivieren der Outputs für SNMP ..... 101
Die wichtigsten Schritte bei SNMP-Zugriff ..... 102
5.4.2 OPC - standardisierter Zugriff ..... 109
Die Installation des OPC-Servers ..... 109
Deinstallation ..... 109
Konfiguration ..... 109
Web-IO als OPC-Device konfigurieren ..... 112
5.4.3 Modbus TCP - standardisierter Zugriff ..... 116
Konfiguration für den Modbus TCP Zugriff ..... 116
Modbus TCP Kommunikation ..... 119
Function Code 0x01 Read Coils ..... 121
Function Code 0x02 Read Discrete Inputs ..... 123
Function Code 0x03 Read Holding Registers ..... 123
Function Code 0x04 Read Input Registers ..... 125
Function Code 0x05 Write Single Coil ..... 126
Function Code 0x06 Write Single Register ..... 127
Function Code 0x0F Write Multiple Coils ..... 127
Function Code 0x10 Write Multiple Registers ..... 129
Function Code 0x07 Read Exception State ..... 131
Fehlerhafte Anfragen vom Master ..... 132
Modbus-Adressbereiche für Web-IO Digital ..... 132
Modbus - virtuelle Register ..... 134

5.5 Email und mehr - Alarm bei Zustandsänderung .. 140
5.5.1 Festlegen der Alarmbedingungen ..... 140

5.5.2 E-Mail-Alarm - SMTP ..... 146
5.5.3 SNMP Alarm ..... 150
5.5.4 Alarm über UDP ..... 152
Outputs eines anderen Web-IO über UDP schalten ..... 155
5.5.5 Alarmmeldung an eigene TCP-Applikationen ..... 155
5.5.6 Alarmmeldung an einen TCP-Server ..... 156
Outputs eines anderen Web-IO über TCP schalten ..... 158
5.5.7 Alarmmeldung an einen Syslog-Server ..... 159
5.5.8 Alarmmeldung an einen FTP-Server ..... 162
5.5.9 Allgemeine Informationen zu Alarmen ..... 166

5.6 Zugriff aus der eigenen Anwendung ..... 167
5.6.1 Zugriff aus dem Browser ..... 167
Verwendung von W&T Tags ..... 168
Dynamische Webseiten AJAX-Technik ..... 173
Nutzen des Web-IO eigenen Java-Applets ..... 182
Die Verwendung des Applets mit JavaScript ..... 185
5.6.2 Zugriff aus diversen Hochsprachen ..... 189
Socketprogrammierung mit Kommandosstrings 191
TCP Kommunikation - das Web-IO als Server ..... 193
TCP Kommunikation - das Web-IO als Client ..... 200
UDP-Kommunikation ..... 206
Programmgesteuertes Fehlermanagement ..... 209
BINARY - Socketprogramme mit Binärstrukturen 213
Festlegen des Operationsmodus ..... 214
Das Web-IO als Socket-Server ..... 215
Das Web-IO als Socket-Client ..... 217
Das Web-IO als UDP-Peer ..... 221
Passwort-Schutz ..... 223
BINARY - Die IO-Strukturen ..... 226
Definition der IO-Strukturen ..... 227
5.6.3 Arbeiten mit den IO-Strukturen ..... 228

5.7 Wayback - Schalten per FTP-Login ..... 239
5.7.1 Funktionsprinzip ..... 239
5.7.2 Die Wayback-Kommandos ..... 240
5.7.3 Freischalten des Wayback-Zugriffs ..... 241
5.7.4 Aktivierung der Outputs für Waybac-Zugriff ..... 244

**6 Allgemeine Funktionen 245**

6.1 Time / Date - Uhrzeit und Datum einstellen ..... 246
6.1.1 Zeitzonen ..... 246

6.1.2 Sommerzeit / Winterzeit ..... 247
6.1.3 Beginn und Ende der Sommerzeit ..... 247
6.1.4 Time-Server ..... 248
6.1.5 Manuelles Einstellen von Uhrzeit und Datum ..... 250
6.1.6 Aktivieren aller Time-/Date-Einstellungen ..... 251

6.2 Safety Timeout / State - Zugriffsüberwachung 252
6.2.1 Safety Timeout festlegen ..... 252
6.2.2 Safety State festlegen ..... 253

6.3 Ports - Input- und Output-Konfiguration ..... 255
6.3.1 Input-Konfiguration ..... 255
6.3.2 Output-Konfiguration ..... 257
6.3.3 Logik Funktionen ..... 260
6.3.4 Time Swich ..... 262
6.3.5 Pulsmode ..... 262

6.4 Ändern der gerätespezifischen Anzeigen ..... 265

6.5 Upload/Download ..... 267
6.5.1 Download ..... 267
6.5.2 Upload ..... 268

6.6 Diagnose und Test ..... 271
6.6.1 Fehler-Report ..... 271
6.6.2 Konfiguration ..... 272
6.6.3 LED -Test ..... 276

**7 Anhang 279**

7.1 Alternativen bei der IP-Adressvergabe ..... 280
7.1.1 Vergabe der IP mit Hilfe des ARP-Kommandos ..... 280
7.1.2 Vergabe der IP über die serielle Schnittstelle ..... 281
7.1.3 IP-Vergabe über einen RARP-Server ..... 283
7.1.4 IP-Vergabe per DHCP-Protokoll ..... 284
7.1.5 IP-Vergabe per BOOTP-Protokoll ..... 288

7.2 Firmware Update ..... 290
7.2.1 Wo ist die aktuelle Firmware erhältlich? ..... 290
7.2.2 Firmware-Update über das Netzwerk mit Windows ... 290

7.3 Notzugang ..... 292

7.4 Verwendete Ports und Netzwerksicherheit ..... 293
7.4.1 Verwendete Ports ..... 293

7.4.2 Das W&T Web-IO und die Netzwerksicherheit ..... 293
7.4.3 Das Berechtigungskonzept des Web-IO ..... 294
7.4.4 Ports mit Sonderfunktion ..... 295
7.4.5 Wiederherstellung der Factory Defaults ..... 296

7.5 Timing und Grenzen bei der Signalverarbeitung 298
7.5.1 Signalerkennung ..... 298
7.5.2 Signalübermittlung ..... 298
7.5.3 Signaltiming ..... 299

7.6 Inventarisierung von W&T Geräten ..... 300
7.6.1 Das Boxinfopaket ..... 300
7.6.2 HIT - Hardware Identification Table ..... 304

7.7 Fehlermeldungen ..... 308

7.8 Technische Daten ..... 310
7.8.1 Technische Daten 57630 / 57631 ..... 310
7.8.2 Technische Daten 57637, 57638 ..... 311
7.8.4 Technische Daten 57634 ..... 312

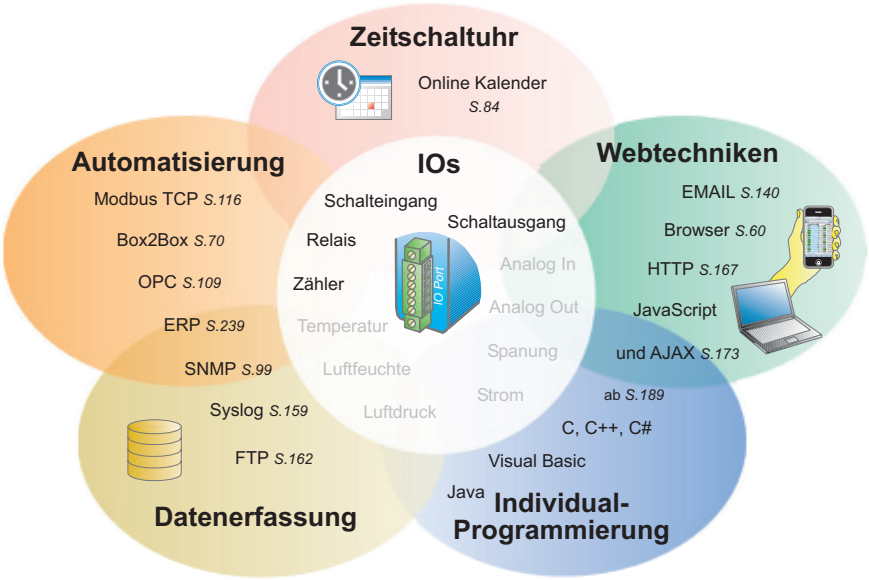
7.9 Entsorgung ..... 313

Index ..... 314

# 1. Übersicht der Einsatzmöglichkeiten

Die Geräte der W&T Web-IO Digital Produktfamilie stellt dem Anwender eine modellabhängige Anzahl von Schaltein- und ausgängen auf Schraubklemme zur Verfügung.

Über eine TCP/IP-Ethernet 10/100BaseT Schnittstelle erlauben die Web-IO Boxen den Zugriff über eine Vielzahl von Protokollen und Anwendungen.



## 1.1 Übersicht der Einsatzmöglichkeiten

Bei entsprechender Konfiguration kann auf die Outputs, Inputs und Counter der Web-IO Boxen auf verschiedensten Wegen zugegriffen werden.

### 1.1.1 Browser-Zugriff

Im einfachsten Fall kann auf die Inputs und Outputs des Web-IO direkt aus dem Browser zugegriffen werden. Das Web-IO bietet dazu fertige Webseiten:

**für PC-Browser**
Die Home- und die User-Seite zeigen selbstaktualisierend die Status von Outputs, Inputs und Countern. Auch das Schalten der Outputs und Löschen der Counter ist möglich.

**für das Smartphone**
Die Smart-Seite gibt einen sehr kompakten, auf die Displaygröße von Mobiltelefonen optimierten Überblick über die Status von Outputs, Inputs und Countern. Außerdem können die Outputs geschaltet werden.

**Eigene Webseiten**
Natürlich ist es auch möglich, die IOs und Counter des Web-IO in eigenen Webseiten anzuzeigen bzw. die vorbereiteten Webseiten durch eigene zu ersetzen.

### 1.1.2 Box-to-Box

In dieser Betriebsart folgen die Outputs eines Web-IO den Inputs eines anderen. Auf diese Weise können Schaltsignale zwischen zwei Standorten über das Netzwerk übertragen werden.

### 1.1.3 Zeitgesteuertes Schalten

Die Outputs des Web-IO können zeitgesteuert geschaltet werden. Die Schaltzeiten lassen sich über 2 Wege konfigurieren:

**Nutzung eines lokalen Kalenderprogramms**
Programme wie z.B. Outlook oder Sunbird bieten die Möglichkeit, Termine (die Schaltzeiten werden wie Termine behandelt) als iCal-Datei zu exportieren. Das Web-IO kann diese iCal-Dateien importieren und in Schaltzeiten umrechnen.

**Schaltzeiten im Google Kalender planen**
Wie normale Termine können die Schaltzeiten im Google Online Kalender festgelegt werden. Das Web-IO ruft die Kalenderdaten danach automatisch, wenn gewünscht zyklisch, bei Google ab und berechnet entsprechend die Schaltzeiten.

### 1.1.4 Integration in bestehende Systeme

Für eine einfache Integration in bestehende Systeme unterstützt das Web-IO die wichtigsten Kommunikationsstandards.

**OPC**
Auf der beiliegenden Produkt CD wird der kostenlose OPC-Server für W&T-Geräte mitgeliefert. Damit können Web-IO in alle OPC-fähigen Visualisierungs- und Leitsysteme eingebunden werden.

**Modbus TCP**
Das Web-IO arbeitet bei Bedarf als Modbus TCP Client und stellt die Zustände von Outputs, Inputs und Countern über entsprechende Speicherbereiche zur Verfügung.

**SNMP - Simple Network Management Protocol**
Durch die Unterstützung von SNMP kann das Web-IO sehr einfach Schaltzustände an Netzwerkmanagement-Systeme weitergeben. Auch das Versenden von SNMP-Traps ist möglich.

Eine passende MIB kann direkt aus dem Web-IO heruntergeladen werden.

### 1.1.5 Zugriff aus eigenen Anwendungen

Die (Socket-)Schnittstelle des Web-IO bietet drei Wege:

**Kommunikation per HTTP-Requests**
Über HTTP-Requests kann das Web-IO aus Web-Anwendungen mit Techniken wie JavaScript, AJAX und PHP angesprochen werden. So können individuelle Anwendungen für den Browser erstellt werden.

**Kommunikation per Kommandostring**
In diesem Modus wird mit einem lesbaren, an HTTP angelehnten Befehlssatz gearbeitet.

**Kommunikation über Binärstrukturen**
In dieser Betriebsart erfolgt die Kommunikation über binäre Datenstrukturen.

In den letzten beiden Varianten kann das Web-IO als TCP-Client oder Server, aber auch als UDP-Peer arbeiten. Alle Programmiersprachen die eine TCP/IP-Kommunikation zulassen sind geeignet, um Anwendungen für das Web-IO zu programmieren (C, C++, C#, VB, VB.net, JAVA, Delphi, Lazarus, Pearl, Python, ...).

Selbst ein Zugriff aus embedded Systemen und SPS ist möglich.

### 1.1.6 Email und mehr - Info bei Zustandsänderung

Das Web-IO hat ein internes Alarmsystem. Angestoßen durch das Eintreten bestimmter Zustände an den Inputs, Outputs oder Countern kann ein Alarm ausgelöst werden.

Für die Übermittlung der Alarme stehen folgende Wege zur Verfügung:

- Email
- SNMP-Trap
- SYSLOG

- Meldung an TCP-Server
- Meldung per UDP
- FTP

### 1.1.7 Erfassen von Schaltzuständen

**Meldungen über das Web-IO Alarmsystem**
Um die Schaltzustände des Web-IO mit Zeitstempel zu erfassen, kann ebenfalls das Alarmsystem des Web-IO genutzt werden. Als Auslöser kann dazu, neben einer Statusänderung der IOs, ein Timer genutzt werden. Sinnvolle Wege zur Datenspeicherung sind:

- SYSLOG
- Meldung an TCP-Server
- Meldung per UDP
- FTP

Aber auch Email und SNMP-Trap können genutzt werden.

**IO-Zustände in W&T-Motherbox erfassen**
Die W&T-Motherbox ist ein Webserver, der auch eine MySQL-Datenbank bereit stellt, um die Daten von Web-IO Produkten zu speichern.

Es kann zwischen zwei Aufzeichnungsmodellen gewählt werden:

- Die Motherbox fragt die beteiligten Web-IO zyklisch ab und speichert die gewonnenen Daten in einem festen Zeitraster.
- Die Web-IOs übermitteln ihre Daten bei Zustandsänderung an die Motherbox.

### 1.1.8 Logik Funktionen - Inputs und Outputs verknüpfen

Eine weitere Möglichkeit, die Outputs zu schalten ist eine logische Verknüpfung zwischen Inputs und Outputs. Der Zustand eines Outputs kann in Abhängigkeit eines definierten Inputmusters auf ON oder OFF geschaltet werden.

### 1.1.9 Basisverhalten der Ports

**Inputs**
Die Inputs sind für 24V Automatisierungssignale vorbereitet, können aber mit Spannungen zwischen -30V und + 30V beschaltet werden. Ein ON-Signal wird ab 8V +/-1V sicher erkannt.

Über einen konfigurierbaren zeitlichen Filter können Kontaktprellen und Störpulse ausgeblendet werden.

**Outputs**
Die Modelle 57630M, 57631M, 57637 und 57636 haben stromtreibende Outputs, die mit jeweils 500mA belastet werden können. Um höhere Ströme zu treiben, können bis zu vier Outputs zu einer gemeinsam schaltenden Gruppe zusammengefasst werden. Das Modell 57634M Schaltet die Outputs mit max. 5A über Relais.

Die Outputs können so konfiguriert werden, dass sie nach dem Einschalten nach einer vorgegebenen Zeit wieder in Ruhe Fallen (Pulsmode)

### 1.1.10 Allgemeine Basisfunktion

Neben den Funktionen des Web-IO, die direkt auf Inputs und Outputs zugreifen, gibt es noch einige wichtige Systemfunktionen. Dazu gehören:

- Web-Based Management
- Systemuhr inkl. Timeserver-Synchronisation
- Möglichkeiten zum Firmware Update

*Alle hier kurz vorgestellten Funktionen sind in den nachfolgenden Kapiteln noch einmal ausführlich beschrieben.*

# 2 Vorbereitungen zur Inbetriebnahme

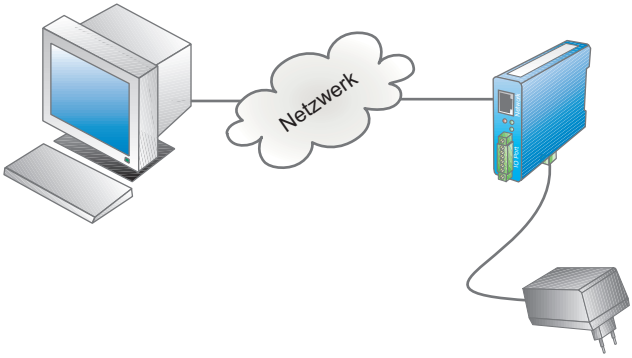
Wir empfehlen in jedem Fall, vor Inbetriebnahme des Web-IO Digital die komplette Anleitung zu lesen.

- Anschluss der Versorgungsspannung
- Verbinden mit dem Netzwerk
- Vergabe einer IP-Adresse

## 2.1 Versorgungsspannung und Netzwerkanschluss

### 2.1.1 Web-IO #57637 und #57638

Verbinden Sie das Web-IO über ein Patchkabel mit einem freien HUB- ober SWITCH-Port. Das Web-IO hat einen autosensing Netzwerkanschluss 10/100BaseT und erkennt automatisch die Netzwerkgeschwindigkeit.



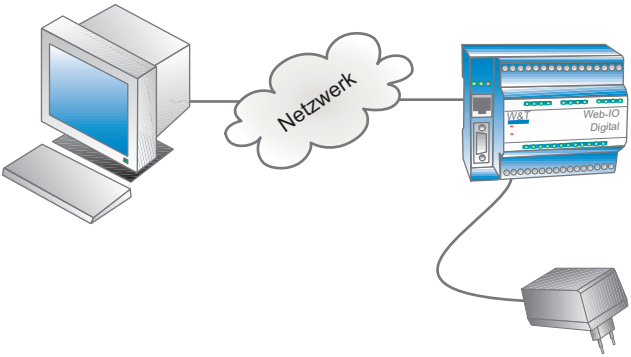

Wenn das Web-IO nicht an einem PoE fähigen Switch angeschlossen ist, klemmen Sie eine Versorgungsspannung zwischen 24 V und 48 V AC oder DC an die Versorgungsspannungsklemme an der Unterseite des Web-IO.

Nach ca. 2 Sek. sollte die Status-LED blinken und die Error-LED erloschen sein. Sollte die rote Error-LED blinken, überprüfen Sie die Verbindung zum Netzwerk.

*Passende Netzteile können bei W&T als Zubehör bezogen werden. Siehe* http://www.wut.de *Bereich* Web-IO Digital

### 2.1.2 Web-IO #57630, #57631, #57634 und #57636

Verbinden Sie das Web-IO über ein Patchkabel mit einem freien HUB- ober SWITCH-Port. Das Web-IO hat einen autosensing Netzwerkanschluss 10/100BaseT und erkennt automatisch die Netzwerkgeschwindigkeit.




Klemmen Sie eine Versorgungsspannung zwischen 12 V und 24 V AC oder DC an die Anschlüsse 1 und 2.

Nach ca. 2 Sek. ertönt ein Piepton und es sollten die Status- und die Error-LED erloschen sein. Sollte die rote Error-LED blinken, überprüfen Sie die Verbindung zum Netzwerk.

*Hat das Web-IO keine bzw. die IP-Adresse 0.0.0.0, bleiben die LEDs* on error *und* system error *nach einem Reset oder Neustart an! Erst wenn eine IP-Adresse vergeben wird, gehen die LEDs aus.*

## 2.2 Vergabe der IP-Adresse

Das Web-IO hat ab Werk die IP-Adresse 0.0.0.0. Vor der Vergabe müssen Sie von Ihrem jeweiligen Systembetreuer eine zu Ihrem Netzwerk passende IP-Adresse erhalten. In kleineren, ungerouteten Netzen verwenden Sie die IP-Adresse Ihres PCs und ändern lediglich die letzte Stelle.

*Jede IP-Adresse muss immer netzwerkweit eindeutig sein.*

### 2.2.1 IP-Adresse automatisch per DHCP beziehen

Ab Werk ist das Web-IO so vorkonfiguriert, dass es beim Systemstart automatisch versucht eine IP-Adresse von einem DHCP-Server zu beziehen. Dabei werden als weitere Parameter übergeben:

- Subnet-Mask
- Gateway-Adresse
- DNS-Server
- Lease-Time

*Bitte beachten Sie, dass dem Web-IO ggf. bei Neustart eine andere IP-Adresse zugeteilt wird. Um das Web-IO gezielt ansprechen zu können, sollte deshalb im DHCP-Server eine Adressreservierung oder eine feste Zuordnung zwischen Gerätenamen und vergebener IP-Adresse vorgenommen werden.*

Weitere Details zu DHCP finden Sie im Anhang im Kapitel *Alternative IP-Adressvergabe.*

### 2.2.2 Vergabe der IP-Adresse mit dem Wutility-Tool

*Diese Methode zur Adressvergabe kann nur auf PC mit Windows-Betriebssystemen durchgeführt werden. Sollte Ihnen kein Windows-System zur Verfügung stehen, beachten Sie bitte im Anhang das Kapitel* Alternative IP-Adressvergabe.

Die Wutility-Installationsdateien finden Sie auf der Produkt-CD, in dem Sie nach CD-Start die Artikelnummer Ihres Web-IO in das *Suche*-Feld eingeben.

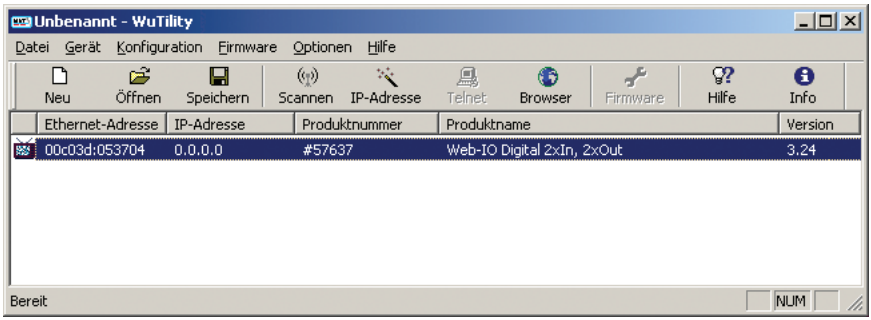
*Alternativ kann Wutility unter www.WuT.de heruntergeladen werden (über* Artikelsuche *und* Tools*).*

Nach erfolgreicher Installation und Start scannt Wutility das Netzwerk nach W&T Geräten ab und listet alle gefundenen Geräte auf.

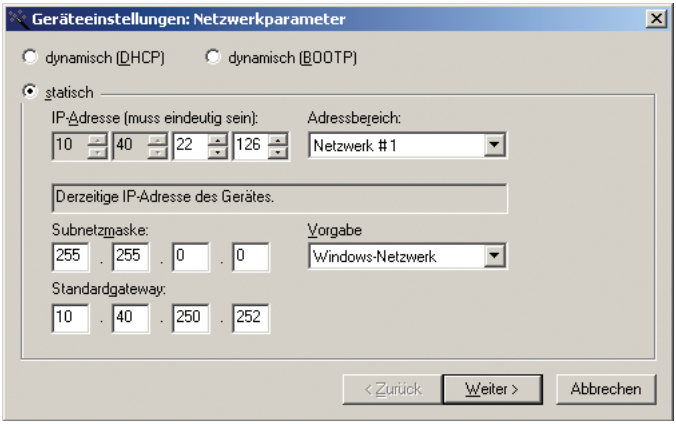
Die Liste der gefundenen W&T Netzwerkkomponenten sollte nun einen Eintrag für das Web-IO enthalten. Die IP-Adresse sollte mit 0.0.0.0 angezeigt werden, wenn es sich um eine Erstinbetriebnahme handelt. Stellen Sie anhand der angezeigten Ethernet-Adresse sicher, ob der Eintrag Ihrem Web-IO entspricht.

Lesen Sie die Ethernet-Adresse des Web-IO von dem Aufkleber am Gehäuse ab:

Markieren Sie den Eintrag für Ihr Web-IO.




Klicken Sie auf das IP-Adresse Icon und geben Sie die IP-Adresse ein, die das Web-I bekommen soll.




Überprüfen Sie, ob die Vorgaben für Subnet-Maske und Gateway stimmen und bestätigen Sie mit *Weiter*. Wenn alle Einstellungen korrekt übernommen wurden, erscheint das Web-IO nun mit der gewünschten IP-Adresse in der Liste.

Alle weiteren Einstellungen können nun bequem über den Browser per Web-Based Management vorgenommen werden.

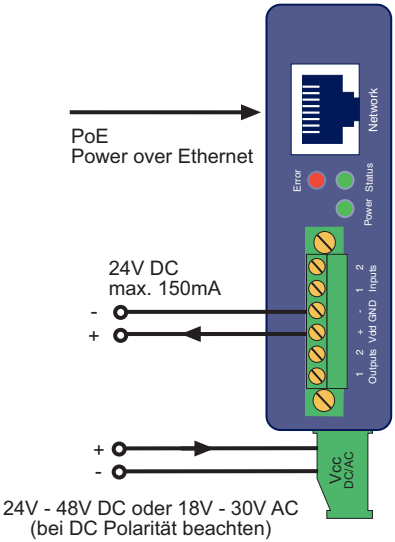
Nutzen Sie nach Login als Administrator den Modus *Profile* um Standardanwendungsfälle zu konfigurieren. Detaillierte Beschreibungen zur Konfiguration der verschiedenen Betriebsarten finden Sie in den nachfolgenden Kapiteln.

# 3 Anschlüsse, Bedienelemente und LEDs

- Versorgungsspannungsanschluss
- Input- und Output-Anschlüsse
- Ethernet-Anschluss
- RS232
- Gerätestatus LEDs

## 3.1 Anschlüsse Web-IO 2x Dig. Input, 2x Dig. Output

**Typ 57637, 57638**




Im Folgenden sind die zwei Möglichkeiten beschrieben, das Web-IO 2x Dig. In, 2x Dig. Out mit Spannung zu versorgen.

*Wird das Gerät via PoE mit der benötigten Betriebsspannung versorgt, kann das Anschließen oder Entfernen einer zusätzlichen externen Spannungsquelle im laufenden Betrieb zu einem Neustart führen.*

### 3.1.1 Externe Spannungsversorgung

Schließen Sie eine Spannungsversorgung von 24V...48V DC (+/-10%) oder 18Veff...30Veff AC (+/-10%) an der Klemme auf der Unterseite des Gerätes an. Sie können hierzu die von W&T angebotenen Netzteile oder alternativ jede beliebige Spannungsversorgung verwenden, welche die technischen Voraussetzungen erfüllt.

*Die externe Spannungsversorgung des Gerätes ist in*

*Netzwerken ohne PoE-Unterstützung immer erforderlich, kann aber auch in PoE-Umgebungen angewendet werden.*

Bei Versorgung mit Gleichspannung muss auf die korrekte Polung geachtet werden. Eine entsprechende Beschriftung ist an den Schraubklemmen auf dem grünen Spannungsversorgungsstecker zu finden.

### 3.1.2 Spannungsversorgung über PoE

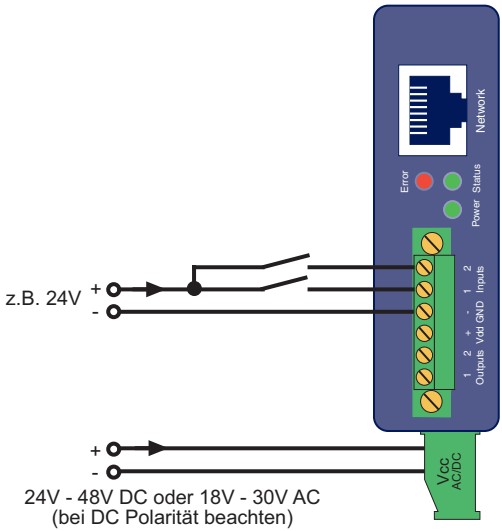
Das Web-IO 2x Dig. In, 2x Dig. Out PoE ist für den Einsatz in Power over Ethernet-Umgebungen gemäß IEEE802.3af ausgerüstet. Die Spannungsversorgung erfolgt hierbei durch die Netzwerkinfrastruktur über den RJ45-Anschluss. Das Gerät unterstützt sowohl die Phantom-Speisung über die Datenpaare 1/2 und 3/6, wie auch die Spare-Pair-Speisung über die ungenutzten Adernpaare 4/5 und 7/8.

Um der versorgenden Komponente ein Powermanagement zu ermöglichen, identifiziert sich das Web-IO als Gerät der Leistungsklasse 1 mit einer Leistungsaufnahme von 0,44W bis 3,84W.

### 3.1.3 Input-Beschaltung

Der erlaubte Eingangsspannungsbereich liegt bei +/-30V gegen die Bezugsmasse.

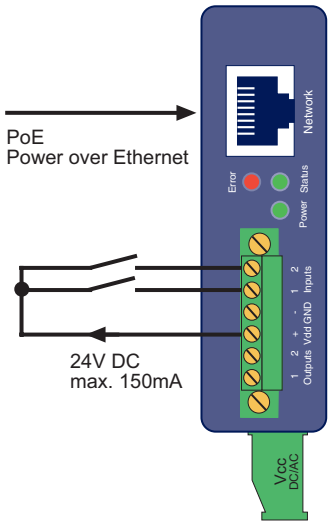
Die Schaltschwelle der Inputs liegt bei 8V +/-1V. Spannungen darunter werden als OFF bzw. 0 Signal erkannt. Spannungen über 8V wertet das Web-IO als ON bzw. 1 Signal. Eingangsspannungen zwischen 7V und 9V sollten vermieden werden, da eine definierte Zuordnung nicht sicher möglich ist.




Das Anschlussbeispiel zeigt die Ansteuerung von zwei Inputs mit Signalen aus verschiedenen Stromkreisen ohne PoE-Versorgung. Dabei ist es wichtig, dass die Input Signale den gleichen Massebezug haben.

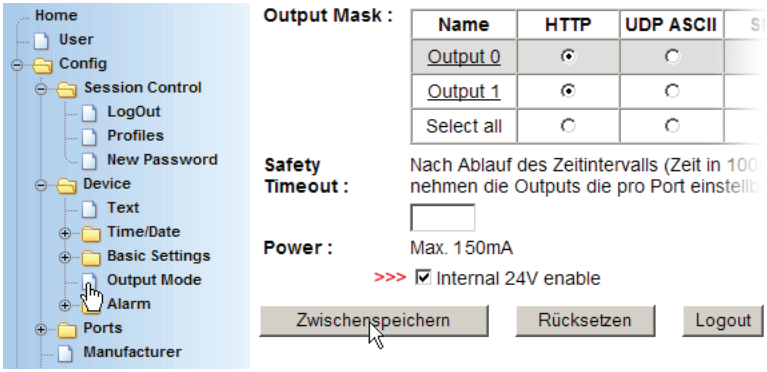
Alternativ besteht die Möglichkeit, das Gerät über PoE mit Strom zu versorgen. In diesem Fall kann die interne Hilfsversorgung genutzt werden, um z.B. ein Signal über potentialfreie Kontakte auf die Inputs zu schalten.

Auf den Klemmen 3 und 4 der Front-Klemmleiste stellt das Web-IO 2x Dig. In, 2x Dig. Out PoE  24V DC zur Verfügung, die mit maximal 150mA belastbar sind.




Die Hilfsspannung muss über das Web-based Management zunächst aktiviert werden.

Loggen Sie sich dazu als Administrator ein und wählen Sie im Menübaum *Config >> Device >> Output Mode*




Aktivieren Sie unter *Power > Internal 24V enable*

Abschließend muss die geänderte Einstellung durch *Logout* und *Speichern* ins Web-IO übernommen werden.

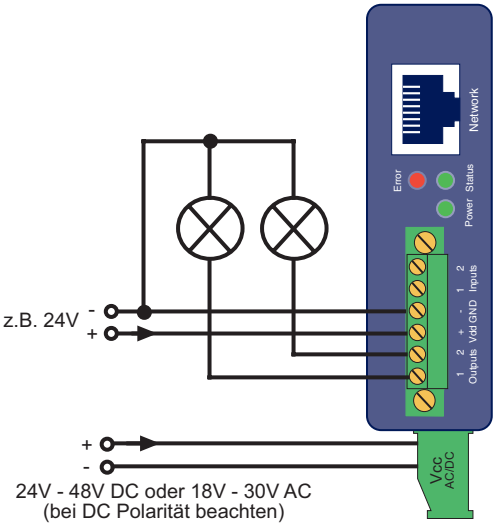
Die Zuleitungen der Inputs sollten nicht länger als nötig sein. Bei Kabellängen unter 2m können ungeschirmte Kabel ge-

nutzt werden.

Bei längeren Zuleitungen sollten in jedem Fall geeignete Schirmmaßnahmen gegen Einstrahlungen getroffen werden. Wir empfehlen die Verwendung geschirmter Kabel und/oder Ferrite, sowie getrennte Kabelführungen, damit Störsignale nicht als Nutzsignale erkannt werden.

Neben der Erkennung des Input-Status (ON/OFF), verfügt jeder Eingang zusätzlich über einen Zähler. Im Auslieferungszustand werden Flanken (Zustandswechsel) gezählt. Die Zähler können alternativ von Flanken- auf Pulszählung umkonfiguriert werden. Darüber hinaus können 2 Inputs zusammen für eine inkrementelle, richtungsabhängige Zählung genutzt werden. Hierbei wird immer der Zähler erhöht, bei dem der Wechsel zuerst erkannt wurde.

### 3.1.4 Output-Beschaltung

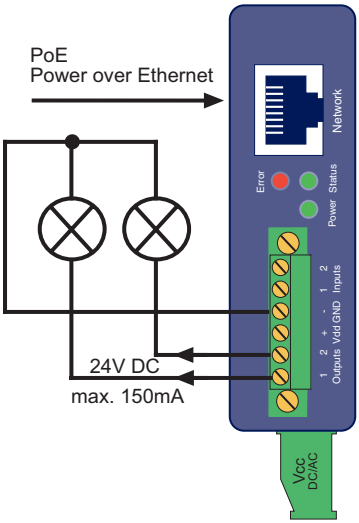



Die 2 Outputs des Web-IO sind stromtreibend. Die Versorgungsspannung für die Outputs kann zwischen +6 V und +30 V (Gleichspannung) liegen und wird über die Front-Klemmen 3 + 4 GND und Vdd eingespeist.

Jeder Output darf max. 500mA schalten. Um höhere Schaltströme zu erreichen, können die beiden Outputs zu einer Gruppe zusammen geschaltet werden. Der maximale Schaltstrom ergibt sich aus der Summe der Einzelströme. Bei 2 Ausgängen max. 1A.

Bei der Dimensionierung der Ausgangsspannungsversorgung sollte der benötigte Strom berücksichtigt werden. Wenn sich die gewählte Output-Spannung zwischen 18V und 30V Gleichspannung bewegt, können Vcc und Vdd auch aus derselben Quelle versorgt werden.

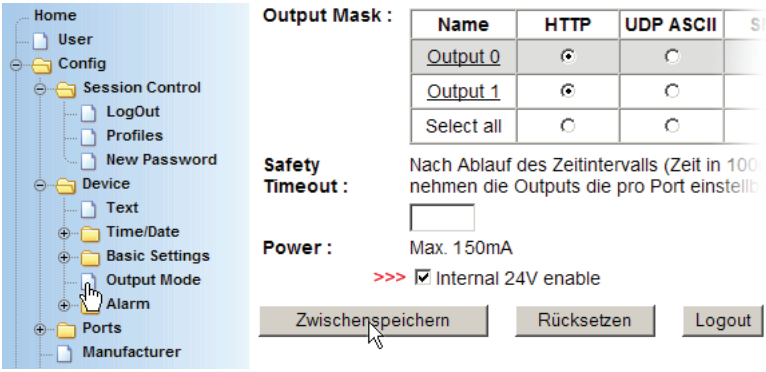
Bei Versorgung über PoE können die Outputs alternativ über die interne 24V Hilfsversorgung gespeist werden.




Für beide Outputs stehen in diesem Fall zusammen max. 150mA Schaltstrom zur Verfügung. Die Hilfsversorgung ist mit einer selbstheilenden thermischen Sicherung abgesichert. Bei Überlast wird die Spannung abgeschaltet und steht nach Entfernen der Last erst nach einigen Minuten wieder zur Verfügung.

Die interne Hilfspannung muss über das Web-Based Management zunächst aktiviert werden.

Loggen Sie sich dazu als Administrator ein und wählen Sie im Menübaum *Config >> Device >> Output Mode.*




Aktivieren Sie unter *Power > Internal 24V enable*

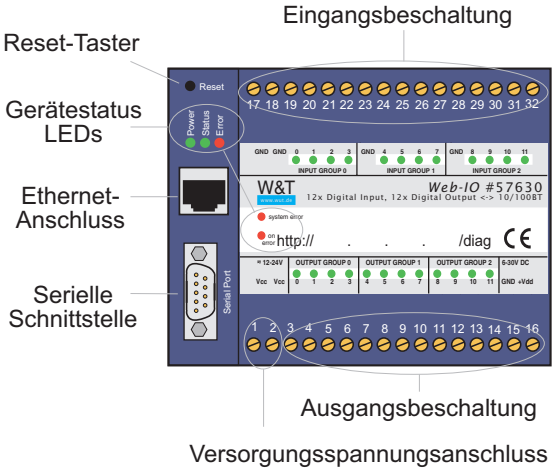
Abschließend muss die geänderte Einstellung durch *Logout* und *Speichern* ins Web-IO übernommen werden.

Outputs, die mit einer induktiven Last wie z.B. einem Relais beschaltet sind, sollten mit einer Freilaufdiode geschützt werden.

Die Outputs verfügen über eine thermische Überlastsicherung und sind kurzschlussfest.

## 3.2 Anschlüsse Web-IO 12x Dig. In, 12xDig. Out

**Type 57630, 57631**




### 3.2.1 Versorgungsspannungseingang

Die Versorgungsspannung wird an den Schraubklemmen 1 und 2 angeschlossen. Die bei W&T erhältlichen Steckernetzteile sind dazu auf der Niederspannungsseite mit einem Spezialstecker ausgerüstet.

Alternativ kann eine vorhandene Spannung zwischen 12V und 24V zur Versorgung angeklemmt werden. Da das Web-IO mit Gleich- oder Wechselspannung betrieben werden kann, spielt die Polarität der Versorgungsspannung keine Rolle.

Um Kabelbruch zu vermeiden, empfehlen wir, die Drahtenden mit Aderendhülsen auszustatten.

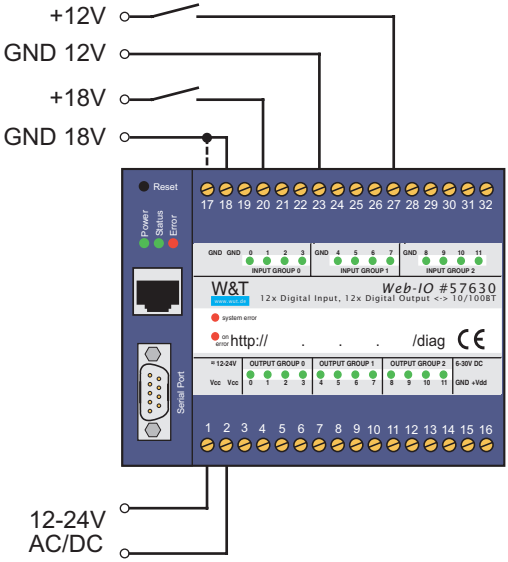
*Die Klemmen 15 und 16 versorgen ausschließlich die Schaltausgänge und benötigen eine Gleichspannung. Bitte lesen Sie hierzu den Abschnitt* Output-Beschaltung.

### 3.2.2 Input-Beschaltung

Die 12 digitalen Eingänge sind in 3 Gruppen zu je 4 Eingängen unterteilt. Die Gruppen sind mit 2kV galvanisch gegeneinander getrennt. Für jede Eingangsgruppe ist eine eigene Bezugsmasse nach außen geführt.

Der erlaubte Eingangsspannungsbereich liegt bei +/-30V gegen die entsprechende Bezugsmasse.

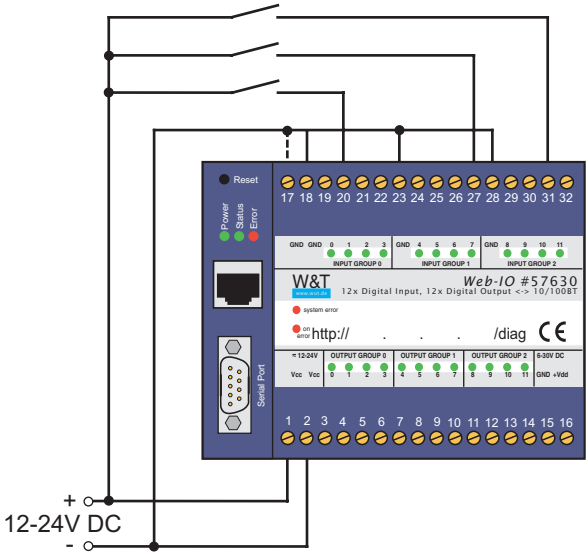
Die Schaltschwelle der Inputs liegt bei 8V +/-1V. Spannungen darunter werden als OFF bzw. 0 Signal erkannt. Spannungen über 8V wertet das Web-IO als ON bzw. 1 Signal. Eingangsspannungen zwischen 7V und 9V sollten vermieden werden, da eine definierte Zuordnung nicht sicher möglich ist.




Das Anschlussbeispiel zeigt die Ansteuerung von zwei Inputs mit Signalen aus verschiedenen Stromkreisen. Dabei ist es wichtig, dass alle Signale einer Input-Gruppe den gleichen Massebezug haben. Bei Signalen mit unterschiedlichem Massebezug sollte die Anschaltung, wie im Beispiel gezeigt, auf getrennte Input-Gruppen erfolgen.

Sollen die Inputs die Zustände potentialfreier Kontakte über-

wachen, kann die Versorgungsspannung Vcc über die Kontakte auf die Inputs geschaltet werden. In diesem Fall muss Vcc mit einer Gleichspannung von 12-24V beschaltet werden.



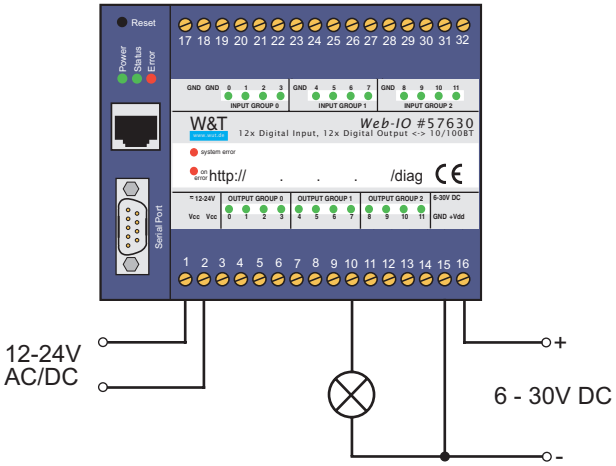

Die Zuleitungen sollten nicht länger als nötig sein. Bei Kabellängen unter 2m können ungeschirmte Kabel genutzt werden.

Bei längeren Zuleitungen sollten in jedem Fall geeignete Schirmmaßnahmen gegen Einstrahlungen getroffen werden. Wir empfehlen die Verwendung geschirmter Kabel und/oder Ferrite, sowie getrennte Kabelführungen, damit Störsignale nicht als Nutzsignale erkannt werden.

Neben der Erkennung des Input-Status (ON/OFF), verfügt jeder Eingang zusätzlich über einen Zähler. Im Auslieferungszustand werden Flanken (Zustandswechsel) gezählt. Die Zähler können alternativ von Flanken- auf Pulszählung umkonfiguriert werden.

### 3.2.3 Output-Beschaltung

Die 12 Outputs des Web-IO sind stromtreibend. Die Versorgungsspannung für die Outputs kann zwischen +6 V und +30 V (Gleichspannung) liegen und wird über die Klemmen 16(Vdd) und 15 (GND) eingespeist.




Jeder Output darf max. 500mA schalten. Um höhere Schaltströme zu erreichen, können einzelne Outputs zu Gruppen von 2 oder 4 zusammen geschaltet werden. Der maximale Schaltstrom ergibt sich aus der Summe der Einzelströme. Bei 2 Ausgängen max. 1A, bei 4 Ausgängen max. 2A.

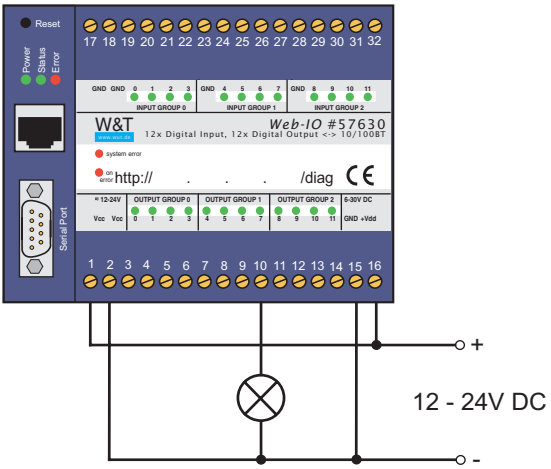
Die Outputs verfügen über eine thermische Überlastsicherung und sind kurzschlussfest. Im OFF-Zustand erfolgt darüber hinaus eine Überprüfung, ob an den Outputs eine Last vorhanden ist. Bei angeschlossener Last fließt daher auch bei abgeschaltetem Output ein Messstrom von ca. 10µA. Das kann bei an den Outputs angeschlossenen LEDs zu einem leichten Glimmen auch im Ruhezustand führen.

Über das Web-based Management kann eine Lastüberwachung aktiviert werden, die den Anwender bei fehlender Last (z.B. Drahtbruch) oder Überlast informiert.

Outputs die mit einer induktiven Last wie z.B. einem Relais

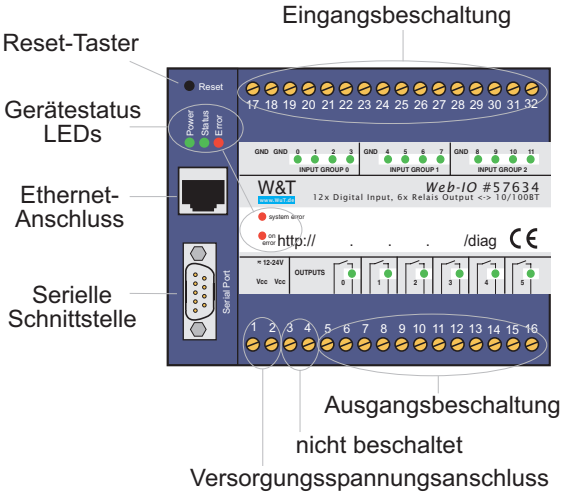
beschaltet sind, sollten mit einer Freilaufdiode geschützt werden.

Bei der Dimensionierung der Ausgangsspannungsversorgung sollte der benötigte Strom berücksichtigt werden. Wenn sich die gewählte Output-Spannung zwischen 12V und 24V Gleichspannung bewegt, können Vcc und Vdd auch aus derselben Quelle versorgt werden.



## 3.3 Anschlüsse Web-IO 12x Dig. Input, 6x Relais

**Type 57634, 57636**




### 3.3.1 Versorgungsspannungseingang

Die Versorgungsspannung wird an den Schraubklemmen 1 und 2 angeschlossen. Die bei W&T erhältlichen Steckernetzteile sind dazu auf der Niederspannungsseite mit einem Spezialstecker ausgerüstet.

Alternativ kann eine vorhandene Spannung zwischen 12V und 24V zur Versorgung angeklemmt werden. Da das Web-IO mit Gleich- oder Wechselspannung betrieben werden kann, spielt die Polarität der Versorgungsspannung keine Rolle.

Um Kabelbruch zu vermeiden, empfehlen wir, die Drahtenden mit Aderendhülsen auszustatten.

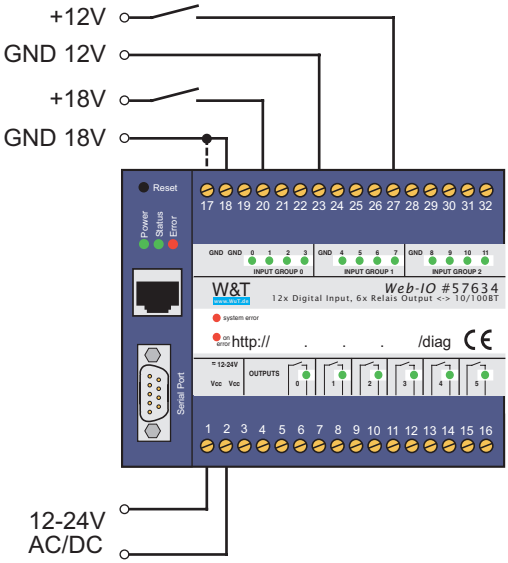
### 3.3.2 Input-Beschaltung

Die 12 digitalen Eingänge sind in 3 Gruppen zu je 4 Eingängen unterteilt. Die Gruppen sind mit 2kV galvanisch gegeneinander getrennt. Für jede Eingangsgruppe ist eine eigene

Bezugsmasse nach außen geführt.

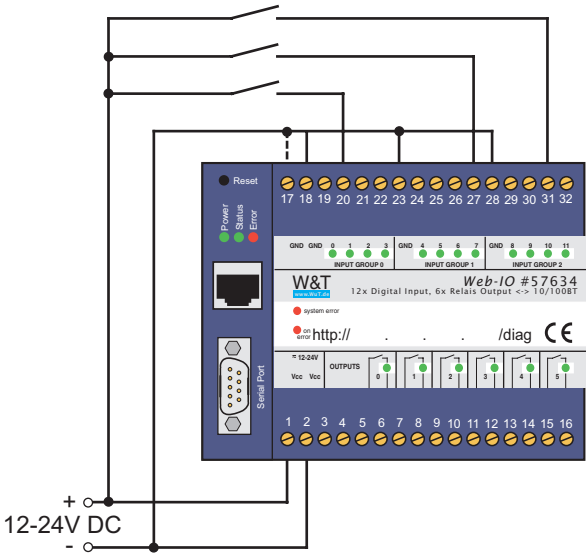
Der erlaubte Eingangsspannungsbereich liegt bei +/-30V gegen die entsprechende Bezugsmasse.

Die Schaltschwelle der Inputs liegt bei 8V +/- 1V. Spannungen darunter werden als OFF bzw. 0 Signal erkannt. Spannungen über 8V wertet das Web-IO als ON bzw. 1 Signal. Eingangsspannungen zwischen 7V und 9V sollten vermieden werden, da eine definierte Zuordnung nicht sicher möglich ist.




Das Anschlussbeispiel zeigt die Ansteuerung von zwei Inputs mit Signalen aus verschiedenen Stromkreisen. Dabei ist es wichtig, dass alle Signale einer Input-Gruppe den gleichen Massebezug haben. Bei Signalen mit unterschiedlichem Massebezug sollte die Anschaltung, wie im Beispiel gezeigt, auf getrennte Input-Gruppen erfolgen.

Sollen die Inputs die Zustände potentialfreier Kontakte überwachen, kann die Versorgungsspannung Vcc über die Kontakte auf die Inputs geschaltet werden. In diesem Fall muss Vcc mit einer Gleichspannung von 12-24V beschaltet werden.



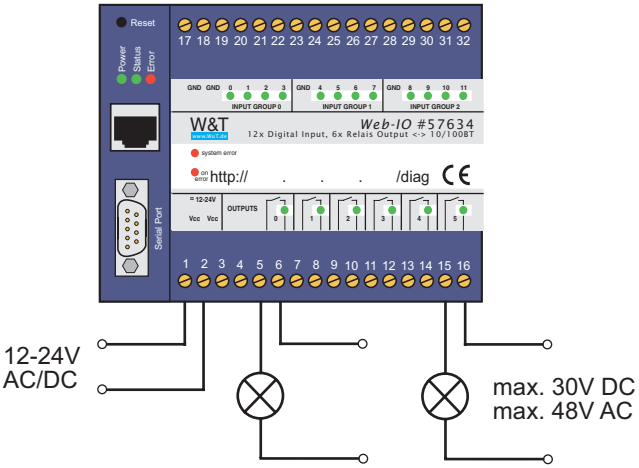

Die Zuleitungen sollten nicht länger als nötig sein. Bei Kabellängen unter 2m können ungeschirmte Kabel genutzt werden.

Bei längeren Zuleitungen sollten in jedem Fall geeignete Schirmmaßnahmen gegen Einstrahlungen getroffen werden. Wir empfehlen die Verwendung geschirmter Kabel und/oder Ferrite, sowie getrennte Kabelführungen, damit Störsignale nicht als Nutzsignale erkannt werden.

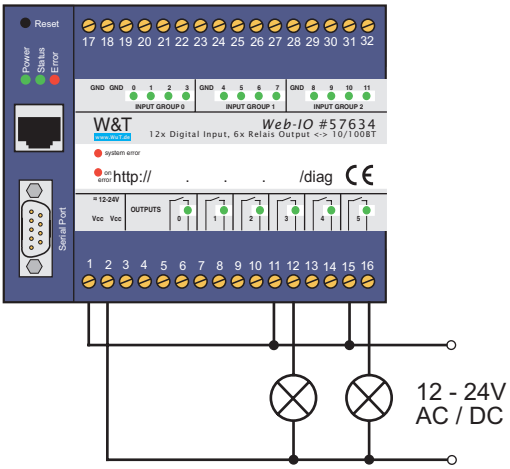
Neben der Erkennung des Input-Status (ON/OFF), verfügt jeder Eingang zusätzlich über einen Zähler. Im Auslieferungszustand werden Flanken (Zustandswechsel) gezählt. Die Zähler können alternativ von Flanken- auf Pulszählung umkonfiguriert werden.

### 3.3.3 Output-Beschaltung

Die 6 Outputs des Web-IO werden intern über Relaiskontakte (Schließer) geschaltet. Die maximale zu schaltende Gleichspannung darf 30V nicht überschreiten wobei die maximale Schaltleistung 150W beträgt. Bei Wechselspannung beträgt die maximale Schaltspannung 48Vbei einer maximalen Last von 240W.

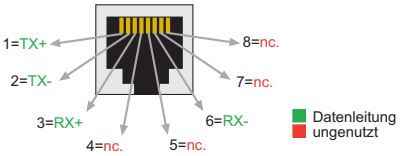



Wenn sich die gewählte Output-Spannung zwischen 12V und 24V bewegt, können das Web-IO und die zu schaltende Last auch aus derselben Quelle versorgt werden.


Reset
Power
Status
Error
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
GND GND 0 1 2 3
INPUT GROUP 0
GND 4 5 6 7
INPUT GROUP 1
GND 8 9 10 11
INPUT GROUP 2
W&T
www.WuT.de
Web-IO #57634
12x Digital Input, 6x Relais Output <-> 10/100BT
system error
error
http:// . . . /diag
CE
≂ 12-24V
Vcc Vcc
OUTPUTS
0 1 2 3 4 5
Serial Port
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
12 - 24V
AC / DC

## 3.4 Ethernet-Anschluss

Die Web-IO Modelle verfügen über einen IEEE 802.3-kompatiblen Netzwerkanschluss auf einem geschirmten RJ45-Steckverbinder. Die Belegung entspricht einer MDI-Schnittstelle, so dass der Anschluss an den Hub oder Switch mit einem 1:1 verdrahteten und geschirmten Patchkabel erfolgt.




*Belegung der RJ45-POE-Netzwerkbuchse*

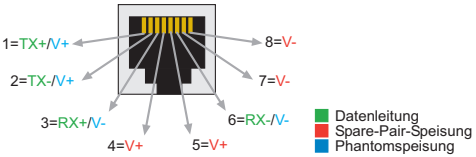
**Auto Negotiation: 10/100BaseT, Full/Half Duplex**

Ab Werk arbeiten alle Web-IO netzwerkseitig in der Betriebsart *Auto-Negotiation*. Datenübertragungsgeschwindigkeit und Duplex-Verfahren werden hierbei mit dem angeschlossenen Switch/Hub automatisch verhandelt und entsprechend eingestellt.

*Beide* Teilnehmer (Switch und Web-IO) sollten in der Betriebsart Auto-Negotiating betrieben werden.

**Galvanische Trennung**

Der Netzwerkanschluss ist sowohl gegenüber der Versorgungsspannung als auch gegenüber den digitalen IOs und der seriellen Schnittstelle mit mindestens $500V_{rms}$ galvanisch getrennt. Web-IO Modelle mit PoE (57637, 57638)




*Belegung der RJ45-POE-Netzwerkbuchse*

Dank der integrierten Power over Ethernet-Technologie, kann das Gerät über den Netzwerkanschluss mit der nötigen Betriebsspannung versorgt werden.

**Link-Status**

Den aktuellen Link-Status zeigt die Error-LED an der Gerätefront an: Blinkt sie im Abstand von 1–2 Sekunden auf, ist die Verbindung zum Hub nicht vorhanden bzw. gestört.

*Managebare Switche verfügen häufig über spezielle Protokolle (Spanning Tree Protocol, Port-Trunking, ...), wie sie z.B. für Uplinks zu anderen Switches oder den breitbandigen Anschluss von Servern benötigt werden. Diese Protokolle werden für die Anbindung gewöhnlicher Endgeräte wie das Web-IO in der Regel nicht benötigt, verzögern den Kommunikationsaufbau nach einem Neustart unter Umständen aber erheblich.*

*In Einzelfällen kann es sogar dazu kommen, dass Switche Daten, die das Web-IO nach Zustandekommen des physikalischen Links sendet, nicht ins Netzwerk weiter geben (z.B. ARP-Anfragen und BootP-Pakete)!*

*Wir empfehlen, die nicht benötigten Protokolle und Funktionen des Switches, an dem für das Web-IO verwendeten Port zu deaktivieren. Wenden Sie sich hierfür bitte an den zuständigen Netzwerkadministrator.*

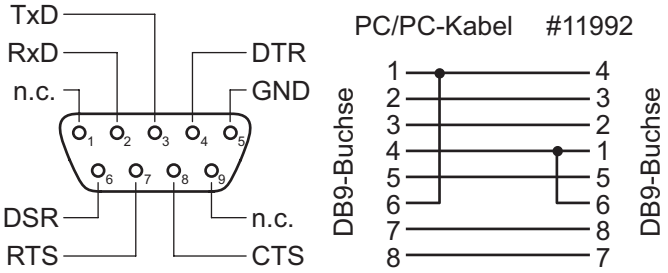
## 3.5 Serieller Anschluss

Beim Web-IO 12x Digital Input, 12x Digital Output ohne Com-Server Funktionalität und beim Web-IO 12x Digital Input, 6x Digital Relais Output steht die serielle Schnittstelle ausschließlich zu Konfigurationszwecken wie z.B. IP-Adressvergabe zur Verfügung.

Die Pinbelegung dieser RS232-Schnittstelle ist identisch mit der eines PCs. Das erlaubt den Einsatz von Standard-Kabeln.

Eine detaillierte Pinbeschreibung sowie die benötigten W&T-Kabel für die wichtigsten Anwendungsfälle finden Sie in den folgenden Abbildungen.

**Pinbelegung und -funktion RS232, DB9 Stecker**




Für die serielle Vergabe der IP-Adresse ist z.B. das Null-Modem-Kabel 1199x geeignet.

*Das Web-IO 2x Digital Input, 2x Digital Output verfügt nicht über eine serielle Schnittstelle.*

## 3.6 LEDs für Gerätestatus

**Power-LED:**
Signalisiert das Anliegen der Versorgungsspannung. Sollte die LED nicht leuchten, überprüfen Sie bitte den korrekten Anschluss der Spannungsversorgung.

**Status-LED:**
Blitzt bei jeglicher Netzwerkaktivität des Web-IO auf. Periodisches Blinken signalisiert, dass der Port Verbindung zu einem anderen Teilnehmer hat.

**Error-LED:**
Die Error-LED weist durch unterschiedliche Blinkcodes auf Fehlerzustände am Gerät oder Netzwerk-Port hin.

**1x Blinken der Error-LED**
Netzwerkanschluss überprüfen. Das Web-IO empfängt keinen Link-Impuls von einem Hub/Switch. Überprüfen Sie das Kabel oder den Hub/SWITCH-Port.

**2x bzw. 3x Blinken der Error-LED**
Führen Sie durch Drücken des Tasters am Gehäuse einen Reset durch (beim Web-IO 2x Digital Input, 2x Digital Output durch Unterbrechung der Spannungsversorgung). Sollte der Fehler nicht behoben sein, setzen Sie das Gerät auf die Factory Defaults zurück. Da alle Netzwerkeinstellungen zurückgesetzt werden, sollten Sie sich Ihre Netzwerkeinstellungen notieren.

Web-Based Management: *Config >> Session Control >> Log-Out >> Restore Defaults* . Nach einem Reset ist das Gerät auf die Factory Defaults zurückgesetzt. Nehmen Sie erneut die Netzwerkeinstellungen vor.

*Mehr Informationen zur Menüführung und Konfiguration finden Sie im Kapitel „Web-Based Management“.*

**Power-LED + Status-LED + Error-LED an**
**= Selbsttest-Fehler**
Der nach jedem Start oder Reset des Web-IO durchgeführte Selbsttest konnte - z.B. wegen eines unvollständigen Updates der Firmware - nicht korrekt beendet werden. Das Gerät ist in diesem Zustand nicht mehr betriebsfähig. Senden Sie das Gerät bitte über Ihren Fachhändler zur Überprüfung an W&T.

**on error http://xxx.xxx.xxx.xxx/diag-LED:**
Zeigt interne Fehler der Konfiguration an. Zur Fehleranalyse rufen Sie die Seite http://<IP-Adresse des Web-IO>/diag im Gerät auf.

Zum Beispiel:

```
http://10.40.22.126/diag
```

**system error:**
Interner Kommunikationsfehler. Versuchen Sie das Gerät durch das Trennen der Spannungsversorgung neu zu starten. Sollte der Zustand anhalten, senden Sie das Gerät bitte über Ihren Fachhändler zur Überprüfung an W&T.

*Hat das Web-IO keine bzw. die IP-Adresse 0.0.0.0, bleiben die LEDs* on error *und* system error *nach einem Reset oder Neustart an! Erst wenn eine IP-Adresse vergeben wird, gehen die LEDs aus.*

*Beim Web-IO 2x Digital Input, 2x Digital Output befinden sich die LEDs im Gehäuseinneren und sind nur von der Seite erkennbar.*

# 4 Konfiguration der Netzwerkparameter

- Web-Based Management
- Grundeinstellung der Netzwerkparameter

## 4.1 Web-Based Management

Das W&T Web-IO Digital stellt dem Anwender ein vollständiges Web-Based Management zur Verfügung. Das bedeutet: Hat das Web-IO eine IP-Adresse, sind alle Einstellungen ganz bequem übers Netzwerk im Browser möglich.

### 4.1.1 Aufbau der Webseite

Wird das Web-IO im Browser aufgerufen, erscheint immer eine zweigeteilte Webseite. Die beiden Teile bezeichnet man auch als Frames.

Auf der linken Seite befindet sich immer der Navigationsframe, den man mit dem Inhaltsverzeichnis eines Buches vergleichen kann.

Rechts ist der Anzeige- und Konfigurationsframe platziert. Beim Aufruf der Webseite ist hier anfangs der aktuelle Status der Inputs und Outputs des Web-IO dargestellt.

### 4.1.2 Die Navigation

Der Navigationsframe beinhaltet einen Verzeichnisbaum, in dem alle verfügbaren Menüpunkte des Web-IO nach Kategorien geordnet aufgelistet sind. Durch Klick werden entweder weitere Menüpunkte einer Kategorie angezeigt und/oder ein neuer Inhalt im Konfigurationsframe aufgebaut.

*Vermeiden Sie die Benutzung der Schaltflächen „Vor“ und „Zurück“ Ihres Browsers. Benutzen Sie ausschließlich den Navigationsbaum bzw. die Buttons und Links im Konfigurationsframe um zu navigieren. Anderenfalls, können im Hintergrund zwischengespeicherte Parameter beim Weiterspringen verloren gehen.*

### 4.1.3 Erleichterte Navigation durch Profile

Das Web-IO Digital bietet diverse verschiedene Betriebsarten und Konfigurationsmöglichkeiten. Um auch dem Anwender der zum ersten mal mit dem Web-IO arbeitet einen einfachen Einstieg zu erlauben, haben wir für die gebräuchlichsten Anwendungsmöglichkeiten Profile erstellt.

Nach erfolgreichem Administrator Login (auf die möglichen Zugriffsrechte geht der nächste Abschnitt ein) hat der Anwender die Wahl zwischen der Konfiguration mit Hilfe der Profile oder dem Expertenmodus, der keine Hilfestellung bei der Konfiguration gibt.

Die Arbeit mit Profilen vereinfacht die Konfiguration durch farbliches Hervorheben der Konfigurationspunkte, die für die entsprechende Betriebsart tatsächlich nötig sind. Dabei werden sowohl die Menüeinträge im Navigationsbaum als auch die jeweiligen Parameter auf den Konfigurationsseiten markiert.

### 4.1.4 Zugriffsrechte

Je nach Login unterscheidet das Web-IO zwischen 3 verschiedenen Zugriffsberechtigungen:

**Default User:**
Diesen Status hat zunächst jeder, der die Startseite des Web-IO aufruft. Der Status von Inputs und Outputs kann ausschließlich angezeigt, aber nicht verändert werden. Änderungen der Konfiguration sind nicht möglich.

**Administrator:**
Durch Klick auf *Config* im Menübaum und Eingabe des Administrator-Passwortes erlangt man den vollständigen Zugriff auf das Web-IO. Die Startseite wird mit einem erheblich erweiterten Menübaum neu aufgebaut. Im Auslieferungszustand ist es nicht nötig, ein Passwort einzugeben und es genügt ein Klick auf den Login-Button.

*Auf die Vergabe der Passwörter wird im nächsten Absatz nä-*

*her eingegangen.*

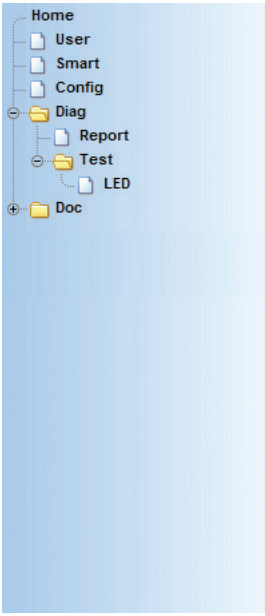
**Operator:**

Die Zugriffsrechte des Operators sind auf das Setzen der Outputs, das Löschen der Counter, das Ändern der Alarmausgaben und einige Diagnosemöglichkeiten begrenzt. Der Operator hat keine weiteren Konfigurationsmöglichkeiten. Das Login in diesen Modus erfolgt ebenfalls durch Eingabe des für diesen Zugriff festgelegten Passwortes.

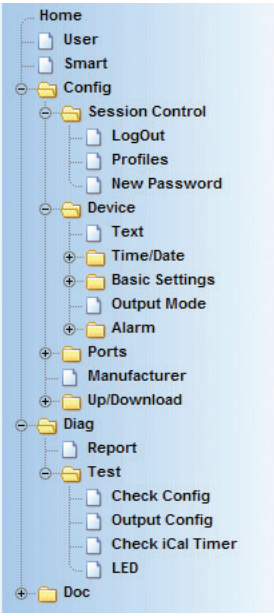
*Das Login erfolgt nach Aufruf der Web-IO Webseite über den Punkt* Config *im Navigationsbaum.*

Hier noch einmal die Menübäume zu den verschiedenen Zugriffsberechtigungen im Vergleich:

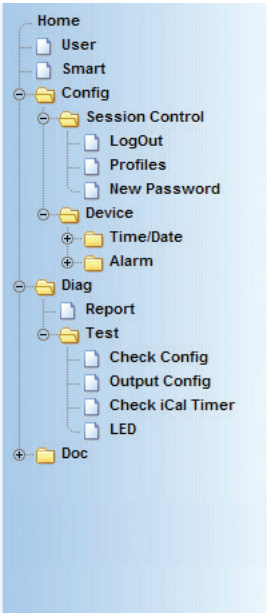
*Default User*




*Administrator*




*Operator*




### 4.1.5 Ändern und Einrichten der Zugriffspasswörter

Die Bedingung hierfür ist das Login als Administrator.

Wählen Sie im Navigationsbaum: *Config >> Session Control >> New Password* und geben Sie in die Eingabefelder die gewünschten Passwörter ein. Die maximal zulässige Länge für ein Passwort ist auf 31 Zeichen begrenzt.

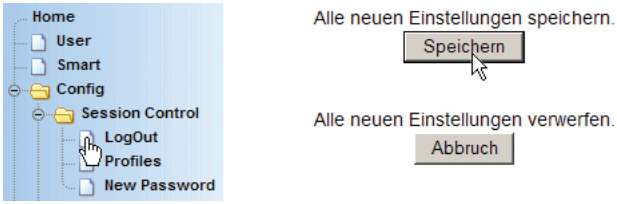
*Es ist nicht zulässig, für Admin- und Operatorzugriff das gleiche Passwort zu benutzen.*

Durch Klick auf den *Zwischenspeichern*-Button werden die neuen Passwörter an das Web-IO übergeben. Nun müssen die Änderungen noch aktiviert werden.

*Tipp: Notieren Sie die vergebenen Passwörter in Ihren persönlichen Unterlagen.*
Sind die Passwörter nicht mehr bekannt, finden Sie im Anhang dieser Anleitung Hilfe.

### 4.1.6 Übernahme und Aktivierung der Einstellungen

Wenn alle Eingaben und Konfigurationsänderungen durchgeführt wurden, müssen die neuen Einstellungen noch aktiviert werden. Benutzen Sie den auf jeder Konfigurationsseite vorhandenen *Logout*-Button oder wählen Sie im Menü Config >> Session Control >> LogOut.




Nach einem Klick auf den Speichern-Button werden alle Ein-

stellungen im Web-IO aktualisiert und die Startseite im Default User Modus neu aufgebaut. Sollen nun weitere Einstellungen geändert werden, muss ein erneutes Login mit den nötigen Zugriffsrechten erfolgen.

*Wird der Browser geschlossen, bevor der* Zwischen-speichern-*Button betätigt wurde, sind die Eingaben unwiederbringlich verloren.*

**Login Timeout**

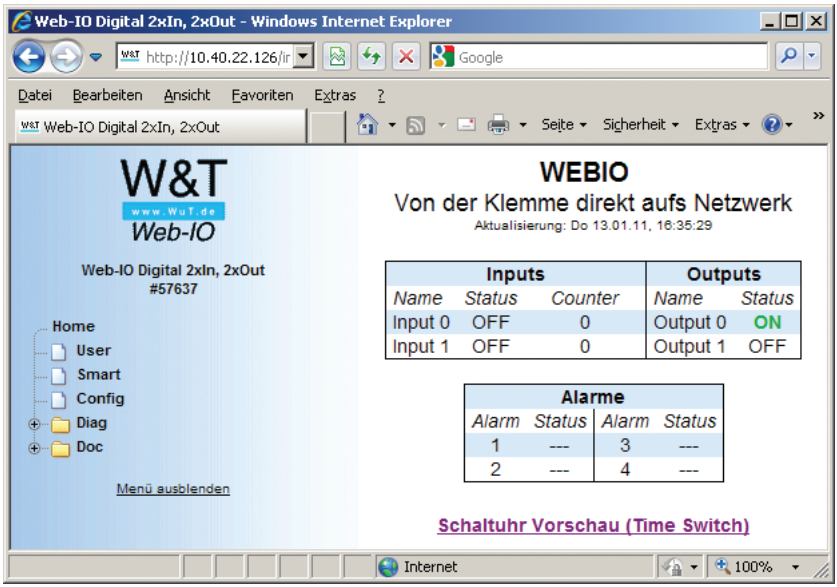
Wird im Administrator- oder Operator-Modus für mehr als 60 Minuten keine Eingabe gemacht, setzt sich das Web-IO auf Default User Rechte zurück. Alle bis dahin gemachten Eingaben und Änderungen, auch wenn sie bereits zum Web-IO gesendet wurden, werden damit verworfen.

## 4.2 Grundeinstellung der Netzwerkparameter

Immer wenn ein Web-IO über die Grenzen des lokalen Netzes hinaus arbeiten soll, werden neben der IP-Adresse einige weitere Netzwerkparameter benötigt.

Um die an dieser Stelle beschriebenen Konfigurationen vornehmen zu können, sollte das Web-IO bereits eine zum Netzwerk passende IP-Adresse haben. Ist das nicht der Fall oder ist die aktuelle IP-Adresse des Web-IO nicht bekannt, führen Sie zunächst die in Kapitel 2 beschriebenen Schritte durch.

Hat das Web-IO bereits eine IP-Adresse, geben Sie diese im URL-Feld des Browsers ein. Nach wenigen Sekunden erscheint die Startseite des Web-IO.




Wählen Sie im Navigationsmenü *Config*.

Sie werden nun aufgefordert, das Administrator-Passwort einzugeben. Im Auslieferungszustand ist kein Passwort vorgegeben und es reicht, den *Login*-Button anzuklicken.

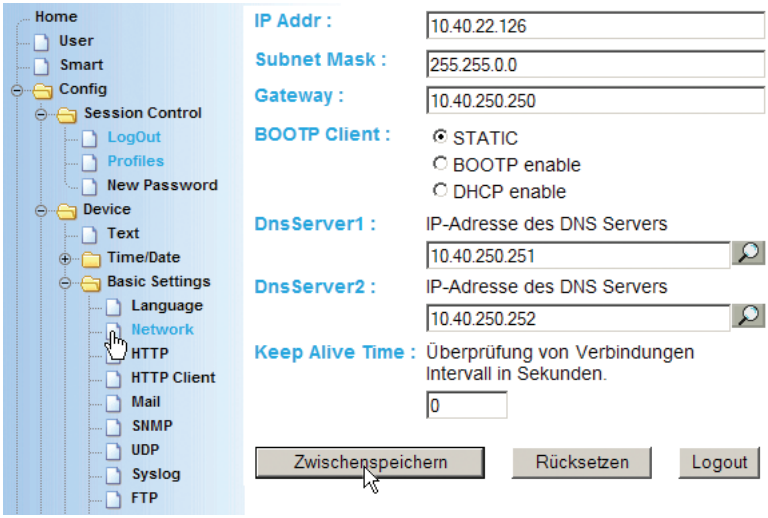
Wurde das richtige Passwort eingegeben, bestätigt das Web-

IO das erfolgreiche Login. An dieser Stelle haben Sie die Auswahl zwischen dem Expertenmodus und dem Arbeiten mit Profilen. Wählen Sie den Button *Profile*.

In der nun erscheinenden Webseite markieren Sie den Radio-Button *Basisparameter Netzwerk* und bestätigen Sie mit Klick auf den *Profile anzeigen*-Button.

Wählen Sie nun im Navigationsmenü den farblich gekennzeichneten Punkt *Config >> Device >> Basic Settings >> Network*.

Es erscheint die folgende Eingabemaske:




Tragen Sie hier die benötigten Netzwerkparameter ein.

**IP Addr.**
Die IP-Adresse kann an dieser Stelle geändert werden.

**Subnet Mask / Gateway**
Für Subnet-übergreifenden Datenaustausch müssen die passende Subnet-Mask und die IP-Adresse des Gateway eingetragen werden.

**DHCP Client / BOOTP Client**
Ab Werk ist das Web-IO so vorkonfiguriert, dass es seine IP-Adresse und weitere, zum Betrieb notwendige Netzwerkparameter von einem DHCP-Server beziehen kann.

In diesem Fall erhält das Web-IO irgendeine IP-Adresse aus dem Adress-Pool des DHCP-Servers, wenn nicht eine spezielle Adressreservierung vorgenommen wurde. Eine gesicherte Nutzung der Server-Dienste des Web-IO ist dann nur möglich, wenn im Netzwerk ein automatisierter Abgleich zwischen DHCP- und DNS-Server stattfindet. Das Web-IO ist dann über den Device Name eindeutig adressierbar.

Alternativ kann das Web-IO über das ältere BootP-Protokoll eine IP-Adresse beziehen, was aber für einen problemlosen Betrieb eine Adressreservierung auf dem DHCP/BootP-Server erfordert.

Soll das Web-IO nicht an der zentralen IP-Adressvergabe via DHCP oder BootP teilnehmen, wählen Sie *static.*

**DNS Server**
Bei einigen Netzteilnehmern, wie z.B. Mail- und Time-Servern, ist es sinnvoll, diese nicht über ihre IP-Adresse sondern über einen Namen zu adressieren. Um Namen in IP-Adressen aufzulösen wird ein DNS-Server benötigt.

Für den Fall, dass DHCP/BootP für die Vergabe der Netzwerkparameter genutzt wird, bekommt das Web-IO auch die Adresse des DNS-Servers automatisch zugewiesen.

Das Feld DNS Server by DHCP wird vom Web-IO ausgefüllt. Änderungen dieses Eintrags durch den Benutzer sind nicht möglich.

Für den manuellen Eintrag von DNS-Servern stehen zwei zusätzliche Felder zur Verfügung.

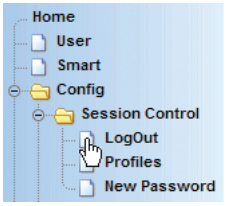
Soll eine IP-Adresse zu einem Namen gesucht werden, wendet sich das Web-IO zuerst an den automatisch zugewiesenen DNS-Server. Ist dieser nicht erreichbar oder fehlt der Ein-

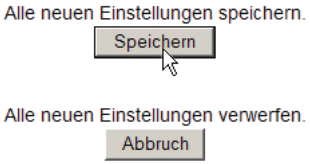
trag, versucht das Web-IO es bei den DNS-Servern 1 und 2.

Mit dem Button kann ein Ping auf den eingetragenen Server ausgelöst werden. So kann überprüft werden, ob der eingetragene Server vom Web-IO erreichbar ist. Wenn der angegebene Server sich nicht im gleichen Subnet befindet, ist die Bedingung hierfür, dass Subnet Mask und Gateway im Web-IO über Logout >> Speichern zuvor abgespeichert wurden.

**Keep Alive Timeout**
In dem hier eingetragenen Intervall überprüft das Web-IO, ob bestehende Verbindungen, auf denen kein Datenverkehr stattfindet noch bestehen. Ist das nicht mehr der Fall, trennt auch das Web-IO die Verbindung um Blockaden zu vermeiden.

Wurden alle Eingaben gemacht, klicken Sie auf den *Zwischenspeichern*-Button. Damit werden die Parameter zunächst in das Web-IO übertragen. Damit sich der Anwender, z.B. bei IP-Adressänderung nicht selbst den Boden unter den Füßen wegzieht, werden die Änderungen erst wirksam, nachdem sie aktiviert werden. Benutzen Sie hierzu den *Logout*-Button oder wählen Sie im Menü Config >> Session Control >> LogOut.




Nach einem Klick auf den *Speichern*-Button wird das Web-IO nun mit den aktuellen Parametern neu gestartet. Zum Neuaufbau der Webseite klicken Sie auf den *hier*-Link.

Beachten Sie, dass Anwendungen, die zu diesem Zeitpunkt mit dem Web-IO kommunizieren, durch den Reset gestört werden können.

Das Web-IO ist nun soweit eingerichtet, dass es auch Subnet-übergreifend angesprochen werden kann.

### Ändern des Device Name

Ab Werk hat jedes Web-IO einen individuellen Gerätenamen, der sich aus der Bezeichnung *WEBIO-* gefolgt von den letzten 3 Stellen der Ethernet Adresse zusammensetzt.

Zum Beispiel lautet der werksseitig eingestellte Device Name eines Web-IO mit der Ethernet-Adresse 00:c0:3d:01:02:03 *WEBIO-010203*.

Der Device Name wird vor allem in Netzwerken benötigt, in denen das Web-IO seine IP-Adresse von einem DHCP-Server bezieht, der einen automatisierten Abgleich mit dem DNS-System vornimmt.

Das Web-IO identifiziert sich innerhalb des DHCP-Protokolls mit seinem Device Name.

Über das Web-Based Management kann der Device Name jederzeit geändert werden.

Notwendige Zugriffsrechte: *Administrator*

Wählen Sie nun im Navigationsmenü den Punkt *Config >> Device >> Text*.

Es erscheint die folgende Eingabemaske:

**Device Name**
System Name mit dem sich das Web-IO über das DHCP-Protokoll bei einem DHCP-Server meldet und der des Weiteren auf der Home-Seite angezeigt wird.

*In Netzwerken mit DHCP und DNS sollte der hier eingetragene Name einmalig sein!*

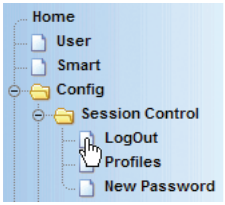
**Device Text**
Der Administrator kann einen Text hinterlegen, der das Web-IO bzw. seine Funktion beschreibt. Dieser Text wird auf der Home-Seite des Web-IO angezeigt.

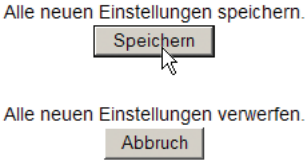
**Location**
Hier kann der Montageort des Web-IO eingetragen werden, um bei größeren Installationen das Auffinden und die Wartung zu erleichtern.

**Contact**
An dieser Stelle kann z.B. eine Emailadresse oder eine Telefonnummer eingetragen werden, unter der im Wartungsfall der zuständige Techniker erreichbar ist.

Zum Speichern der Texte benutzen Sie den *Zwischenspeichern*-Button und wählen Sie im Menü Config >> Session Con-

trol >> LogOut.




Nach einem Klick auf den *Speichern*-Button wird das Web-IO nun mit den aktuellen Parametern neu gestartet.

*Wenn Ihnen die verwendeten Ausdrücke und Bezeichnungen nicht vertraut sind, fordern Sie kostenlos unser Buch* TCP-IP-Ethernet bis Web-IO *an, in dem die nötigen Grundlagen zur Inbetriebnahme von Netzwerkkomponenten vermittelt werden. www.WuT.de*

# 5 Basisfunktionen

- HTTP - Inputs und Outputs im Browser steuern
- Box-to-Box
- Zeitgesteuert schalten
- Integration in bestehende Systeme
- Email und mehr - Info bei Zustandsänderung - Alarme
- Zugriff aus der eigenen Anwendung

## 5.1 HTTP - Inputs und Outputs im Browser steuern

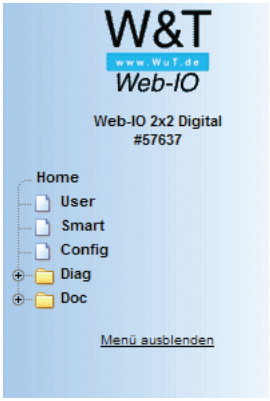
Der Zugriff aus dem Browser ist für den Anwender die einfachste Möglichkeit Inputs sowie Counter zu überwachen und die Outputs zu steuern.

Das Web-IO bietet hierzu 3 Standardansichten bzw. Webseiten:

- Home - Übersicht der IO-Zustände und des Alarmstatus
- User - Übersicht der IO-Zustände und bei Bedarf des freien Modbusspeichers
- Smart - IO-Status optimiert für Smart Phone Anzeige

### 5.1.1 Die Home-Seite

Auf der Home-Seite bekommt der Anwender den Status der Inputs, Counter, Outputs und der konfigurierten Alarme selbstaktualisierend angezeigt.




**WEBIO-05372A**

Von der Klemme direkt aufs Netzwerk

Aktualisierung: Fr 28.01.11, 10:34:13 *

| Inputs | | | Outputs | |
|---|---|---|---|---|
| *Name* | *Status* | *Counter* | *Name* | *Status* |
| Input 0 | OFF | 0 | Output 0 | OFF |
| Input 1 | OFF | 0 | Output 1 | OFF |

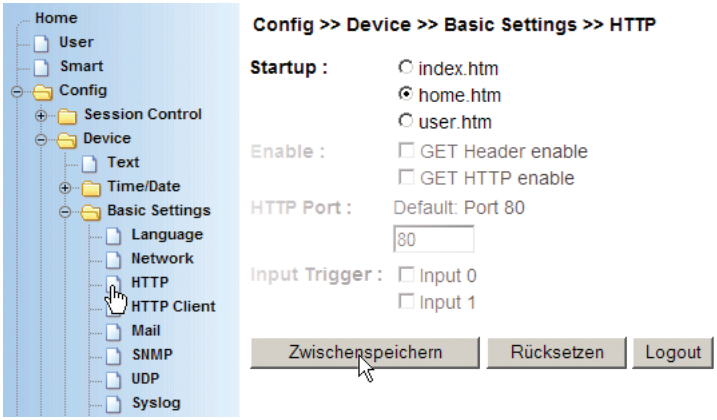
| Alarme | | | |
|---|---|---|---|
| *Alarm* | *Status* | *Alarm* | *Status* |
| 1 | --- | 3 | --- |
| 2 | --- | 4 | --- |

Schaltuhr Vorschau (Time Switch)

Um auch die Outputs bedienen zu können, ist es notwendig, sich über den Navigationsbaum als Administrator oder mit Operator-Rechten in das Web-IO einzuloggen. Ab Werk ist das Web-IO so voreingestellt, dass die Home-Seite neben dem Navigationsbaum zu sehen ist. Um bei Aufruf im Browser nur die Home-Seite ohne Navigationsbaum anzuzeigen gibt es zwei Möglichkeiten:

1. Eingabe in der Adresszeile http://<ip-Adresse>/home
2. Web-IO so konfigurieren, dass nur die Home-Seite angezeigt wird.

   Dazu als Administrator einloggen und unter *Config >> Device >> Basic Settings >> HTTP* bei *Start Up* den Punkt *home.htm* auswählen.




*Alle weiteren Einstellungen auf dieser Seite sind im weiteren Verlauf des Handbuches beschrieben und sollten an dieser Stelle nicht verändert werden.*

*Die Home-Seite kann gegen eine eigene individuell gestaltete Webseite ersetzt werden. Details hierzu im Kapitel „Zugriff aus der eigenen Anwendung“.*

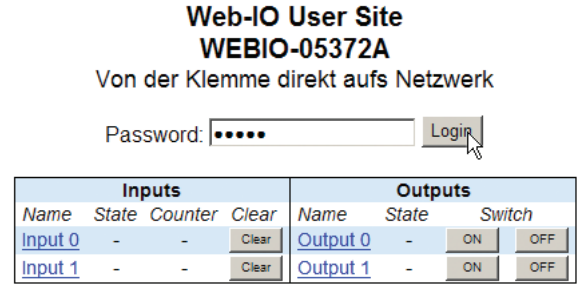
### 5.1.2 Die User-Seite

Die User-Seite zeigt den Status von Inputs, Countern und Outputs. Anders als bei der Home-Seite kann ein Login für die Funktionen der Seite auch ohne Zugriff auf den Navigationsbaum erfolgen. Ein über Login eingetragenes Passwort wird in Form eines Cookies auf dem PC gespeichert.

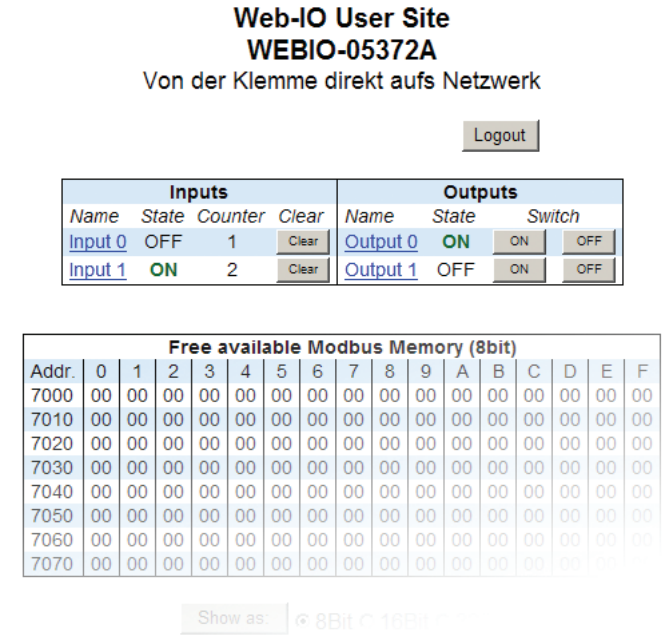
Für den Anwender hat das den Vorteil, dass er vom selben PC aus bei wiederholtem Aufruf der User-Seite, auch ohne erneute Anmeldung auf die Outputs zugreifen kann bzw. die

Anmeldung automatisch erfolgt.

*Wurde für das Web-IO ein Passwort vergeben wird ohne Login der Status von Inputs, Outputs und Countern nicht angezeigt!*




Nach erfolgreicher Passworteingabe werden die Zustände der IOs angezeigt und die Outputs sind bedienbar.



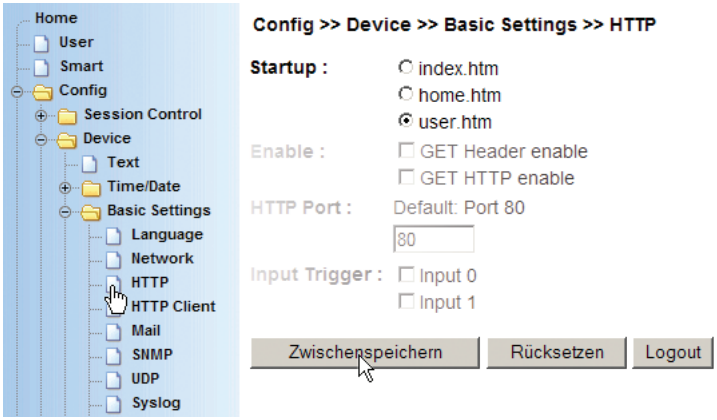

Nur wenn das Web-IO für den Modbus-Zugriff konfiguriert wurde, erscheint unterhalb der IO-Zustände zusätzlich eine

Tabelle, in welcher der Inhalt des frei-verfügbaren Modbus-Speichers visualisiert wird. (Details hierzu im Kapitel „Integration in bestehende Systeme >> Modbus“)

Um bei Aufruf im Browser nur die User-Seite ohne Navigationsbaum anzuzeigen gibt es zwei Möglichkeiten:

1. Eingabe in der Adresszeile http://<ip-Adresse>/user
2. Web-IO so konfigurieren, dass nur die Home-Seite angezeigt wird.

   Dazu als Administrator einloggen und unter *Config >> Device >> Basic Settings >> HTTP* bei *Start Up* den Punkt *user.htm* auswählen.




*Alle weiteren Einstellungen auf dieser Seite sind im weiteren Verlauf des Handbuches beschrieben und sollten an dieser Stelle nicht verändert werden.*

*Auch die User-Seite kann gegen eine eigene individuell gestaltete Webseite ersetzt werden. Details hierzu im Kapitel „Zugriff aus der eigenen Anwendung“.*

### 5.1.3 Die Smart-Seite

Die Smart Seite wurde für die Anzeige in Smartphones wie .B. dem iPhone optimiert und wird über http://<IP-Adresse>/

smart aufgerufen.

Die Smart-Seite zeigt den Status von Inputs, Countern und Outputs. Genauso wie bei der User-Seite kann ein Login für die Funktionen der Seite auch ohne Zugriff auf den Navigationsbaum erfolgen. Ein über Login eingetragenes Passwort wird in Form eines Cookies auf dem Smartphone gespeichert.

Für den Anwender hat das den Vorteil, dass er von seinem Smartphone bei wiederholtem Aufruf der Smart-Seite, auch ohne erneute Anmeldung auf die Outputs zugreifen kann bzw. die Anmeldung automatisch erfolgt.

*Wurde für das Web-IO ein Passwort vergeben wird ohne Login der Status von Inputs, Outputs und Countern nicht angezeigt!*

### 5.1.4 Beschriftungen und Texte

Notwendige Zugriffsrechte: *Administrator*

Das Erscheinungsbild der Web-IO Standardseiten im Brows-

er kann insofern individuell gestaltet und verändert werden, dass die Benennung des Web-IO, der Inputs, Counter und Outputs frei konfigurierbar ist.

**Gerätenamen und Beschreibung**

Im Navigationsbaum unter *Config >> Device >> Text* können Gerätenamen und Beschreibung des Web-IO geändert werden.

Soll sich die Beschreibung über mehr als eine Zeile erstrecken, wird an der für den Zeilenumbruch gewünschten Position das <br> Tag eingefügt.

Die Felder *Location* und *Contact* werden im Browser nicht angezeigt und dienen ausschließlich der Identifikation des Web-IO.

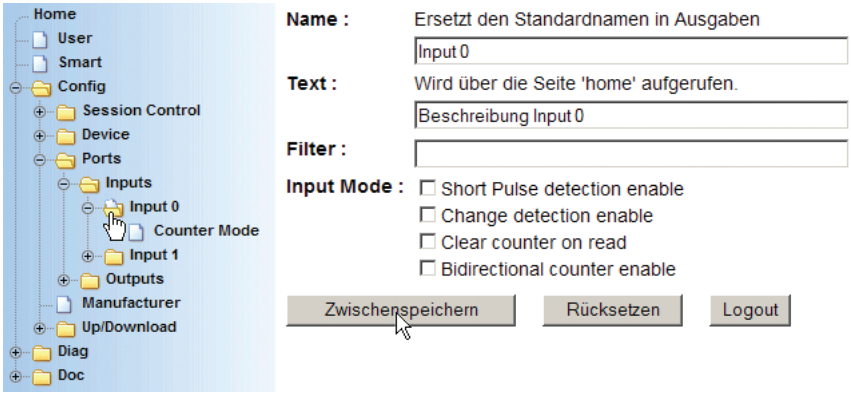
Die Anzeige im Browser ändert sich nach *Zwischenspeichern* und *Logout > Speichern*.

**Port-Bezeichnung und -Beschreibung**

Auch für die Inputs und Outputs können individuelle Bezeichnungen und Beschreibungen eingerichtet werden.

Um die Input-Bezeichnung zu ändern, wählen Sie im Navigati-

onsbaum *Config >> Ports >> Inputs* und für Input 0 *>> Input 0*:




*Die Einstellungen für* Filter *und* Input Mode *lassen Sie bitte unberührt. Detaillierte Informationen zu diesen Parametern erhalten Sie im Kapitel* „Port Einstellungen“

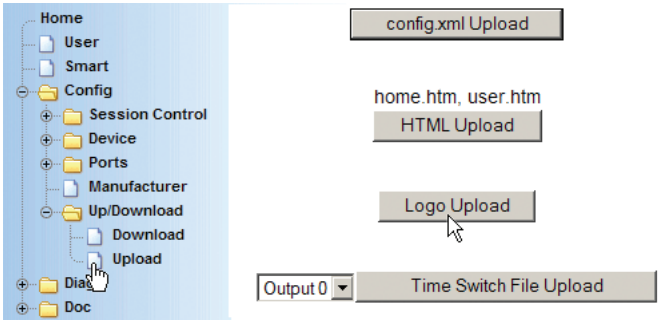
Für Änderungen der Output-Bezeichnungen wählen Sie *Config >> Ports >> Outputs* und z.B. *>> Output 0* für Output 0

*Die Einstellungen für* Group Mode *und* Safety State *lassen Sie bitte unberührt. Detaillierte Informationen zu diesen Parametern erhalten Sie im Kapitel* „Port Einstellungen“

Die Anzeige im Browser ändert sich nach *Zwischenspeichern* und *Logout > Speichern.*

**Logo im Navigations-Frame**

Um ein möglichst kundenspezifisches Erscheinungsbild der Web-IO Startseite zu ermöglichen, besteht die Möglichkeit, das W&T-Logo im Navigations-Frame gegen ein beliebiges andres zu ersetzen. Wählen Sie im Navigationsbaum *Config >> Up/Download >> Upload*

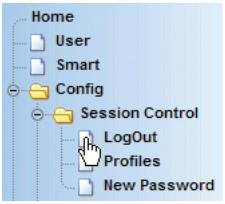


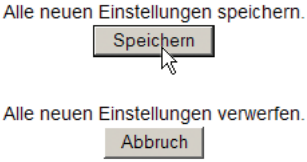

Klicken Sie auf den Button *logo.gif upload*.

Wählen Sie die gewünschte JPG- oder GIF-Datei aus und klicken Sie auf den Button *logo.gif upload.*

Dabei sollte allerdings beachtet werden, dass die Abmessungen des Logos in etwa mit denen des W&T-Logos übereinstimmen (Breite max. 200 Pixel, Höhe max. 100 Pixel), damit die Übersichtlichkeit des Navigations-Frames erhalten bleibt.

Die Anzeige im Browser ändert sich nach *Zwischenspeichern* und *Logout > Speichern*.




*Sollte das neue Logo beim Neustart noch nicht angezeigt werden, löschen Sie bitte den Browser Cache!*

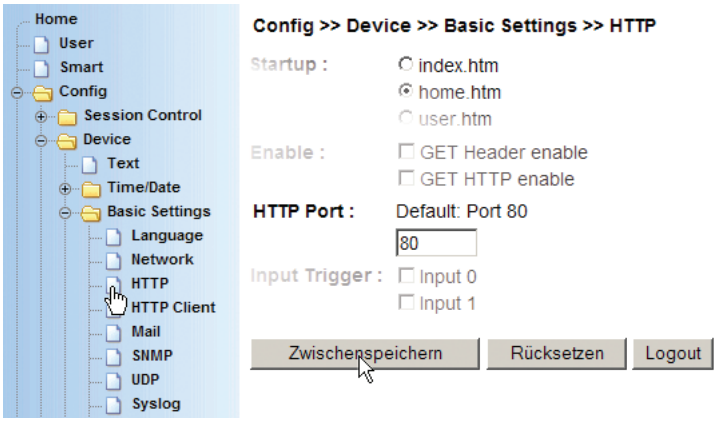
### 5.1.5 Ändern des HTTP-Ports

Der Zugriff des Browsers auf den Webserver des Web-IO erfolgt gemäß HTTP-Protokoll ab Werk über TCP Port 80.

Es gibt Fälle wie NAT-Routing oder Port-Forwarding, die es nötig machen, einen abweichenden Port für den HTTP-Zugriff zu konfigurieren.

Dazu loggen Sie sich als Administrator ein und wählen im Navigationsbaum *Config >> Device >> Basic Settings >> HTTP.*

Unter *HTTP Port* kann ein vom Standard Port 80 abweichender TCP Port für den HTTP-Zugriff eingetragen werden.



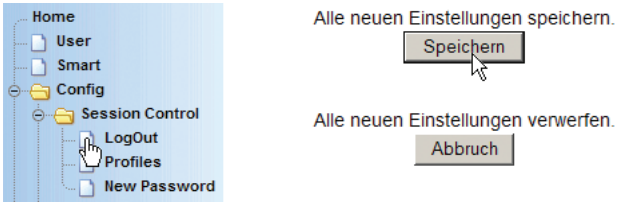

*Alle weiteren Einstellungen auf dieser Seite sind im weiteren Verlauf des Handbuches beschrieben und sollten an dieser Stelle nicht verändert werden.*

Wird der HTTP Port geändert, ist das Web-IO aus dem Browser nur noch über die zusätzliche Angabe des gewählten HTTP Ports erreichbar.

Beispiel für HTTP Port = 85.

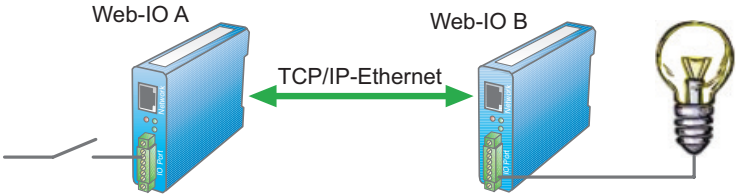
Das gilt auch für alle Konfigurationszugriffe.

Die Änderung wird nach *Zwischenspeichern* und *Logout* > *Speichern* wirksam.



## 5.2 Box-to-Box

In der Betriebsart Box-to-Box werden zwei Web-O so konfiguriert, dass die Outputs von Web-IO B den Inputs von Web-IO A folgen.




So lassen sich z.B. über eine WAN-Anbindung standortübergreifend Signale schalten.

Bei Box-to-Box Verbindungen übernimmt ein Web-IO die Funktion des Masters. Das zweite Web-IO arbeitet als Slave. Der Slave wartet darauf, dass der Master die Verbindung zu ihm aufnimmt.

Unabhängig davon, welche Seite als Master arbeitet und welche als Slave, können Schaltsignale in beide Richtungen weitergegeben werden.

Sowohl das Master, als auch das Slave Web-IO müssen entsprechend konfiguriert werden.

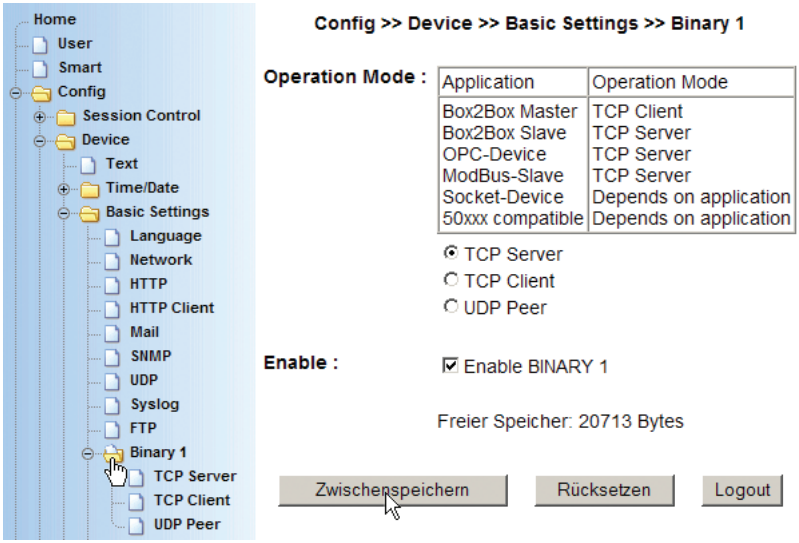
### 5.2.1 Konfiguration des Slave Web-IO

Notwendige Zugriffsrechte: *Administrator*

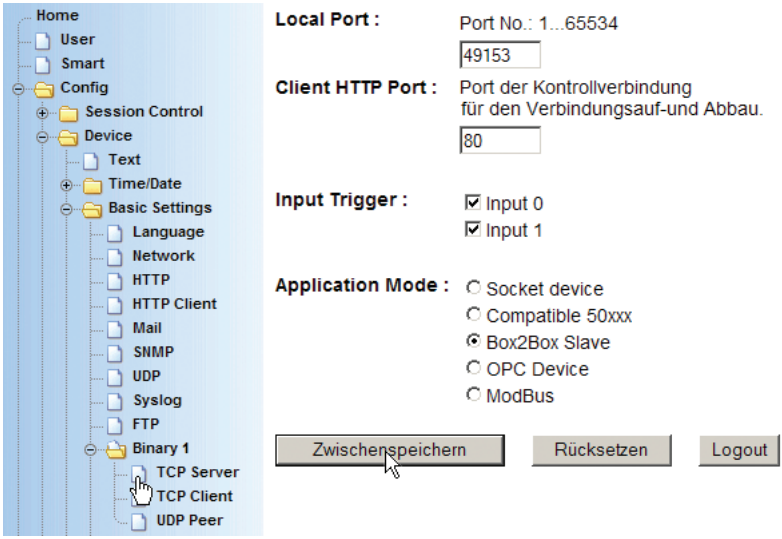
Wählen Sie im Navigationsbaum des Slave Web-IO *Config >> Device >> Basic Settings >> Binary 1*

Stellen Sie als *Operation Mode* die Betriebsart *TCP-Server* ein und aktivieren Sie *Enable Binary*.

Klicken Sie anschießend auf den *Zwischenspeichern*-Button um die Änderungen ans Web-IO zu übertragen.




Wählen Sie nun im Navigationsbaum: *Config >> Device >> Basic Settings >> Binary 1 >> TCP-Server.*




**Local Port:**

Wenn Ihnen von Ihrem Netzwerkadministrator keine besonderen Vorgaben gemacht werden, kann der werksseitig einge-

stellte Port 49153 übernommen werden.

Grund für eine Änderung des werksseitig eingestellten Local Port kann z.B. eine Firewall sein, die nur bestimmte Portzugriffe erlaubt.

*In jedem Fall muss der eingestellte Local Port des Slave identisch mit dem* Server Port *Eintrag des Masters sein.*

**Client HTTP Port**
legt den HTTP Port fest, auf den die Steuerverbindung zur Master-Box aufgebaut werden soll.

Wenn nicht anders vorgegeben, sollte hier immer Port 80 verwendet werden.

**Input Trigger:**
aktivieren Sie hier die Inputs, die die korrespondierenden Outputs beim Master setzen sollen.

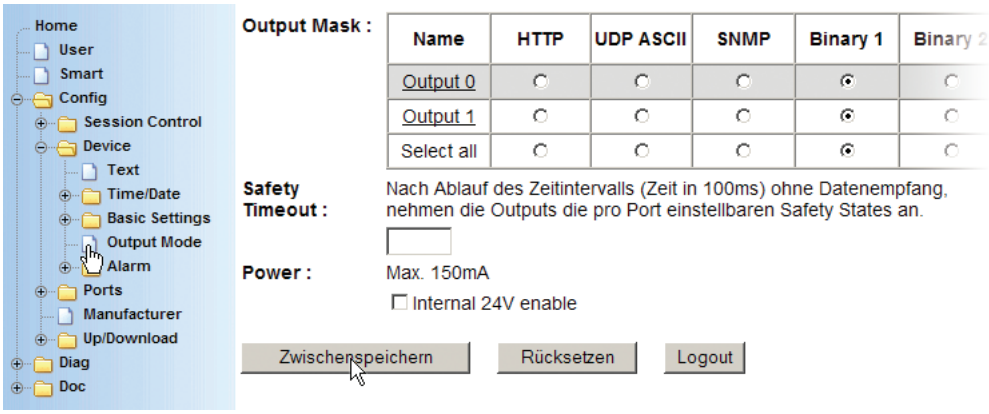
Das Web-IO Digital erlaubt den gleichzeitigen Zugriff auf die Inputs aus verschiedenen Betriebsarten.

So können die Inputs 0-5 z.B. die Outputs 0-5 des Master Web-IO steuern, gleichzeitig aber auch noch über HTTP ausgelesen werden.

**Application Mode**
Markieren Sie *Box2Box Slave*

Nachdem alle Parameter eingegeben wurden, bestätigen Sie durch Klick auf den *Zwischenspeichern*-Button.

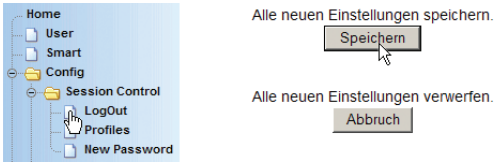
Wählen Sie nun im Navigationsbaum: *Config >> Device >> Output Mode*




Aktivieren Sie hier die Outputs, die von den korrespondierenden Inputs des Slave gesetzt werden sollen für Binary 1 und bestätigen Sie durch Klick auf den *Zwischenspeichern*-Button.

Im Gegensatz zu den Inputs sind die für Box-to-Box aktivierten Outputs für andere Betriebsarten nicht mehr zugänglich.

Nun müssen die neuen Einstellungen noch aktiviert werden. Benutzen Sie den *Logout*-Button oder wählen Sie im Menü *Config >> Session Control >> LogOut.*



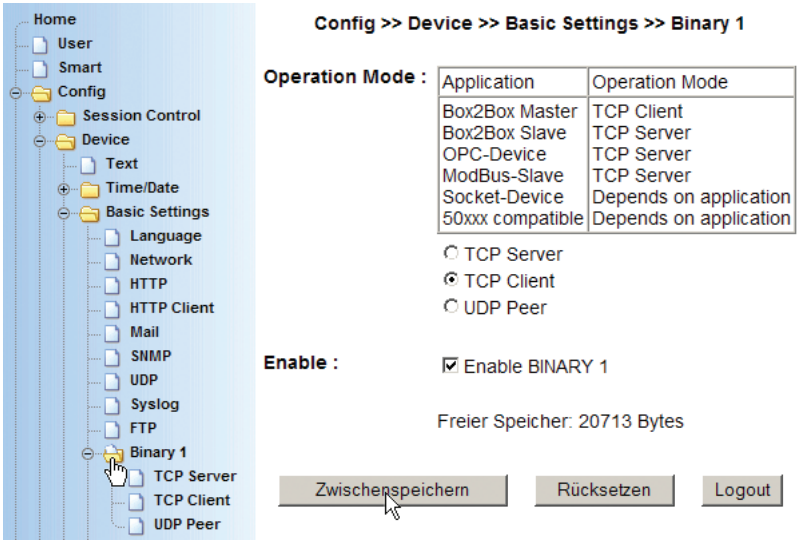

Nach Klick auf den *Speichern*-Button werden alle Einstellungen im Web-IO aktualisiert und die Startseite im Default User Modus neu aufgebaut. Das Master Web-IO kann sich nun mit dem Slave Web-IO verbinden.

*Der Box-to-Box Master kann sich nur dann mit dem Slave verbinden, wenn beim Slave kein HTTP-Login als Operator oder Administrator besteht*

### 5.2.2 Konfiguration des Masters

Notwendige Zugriffsrechte: *Administrator*

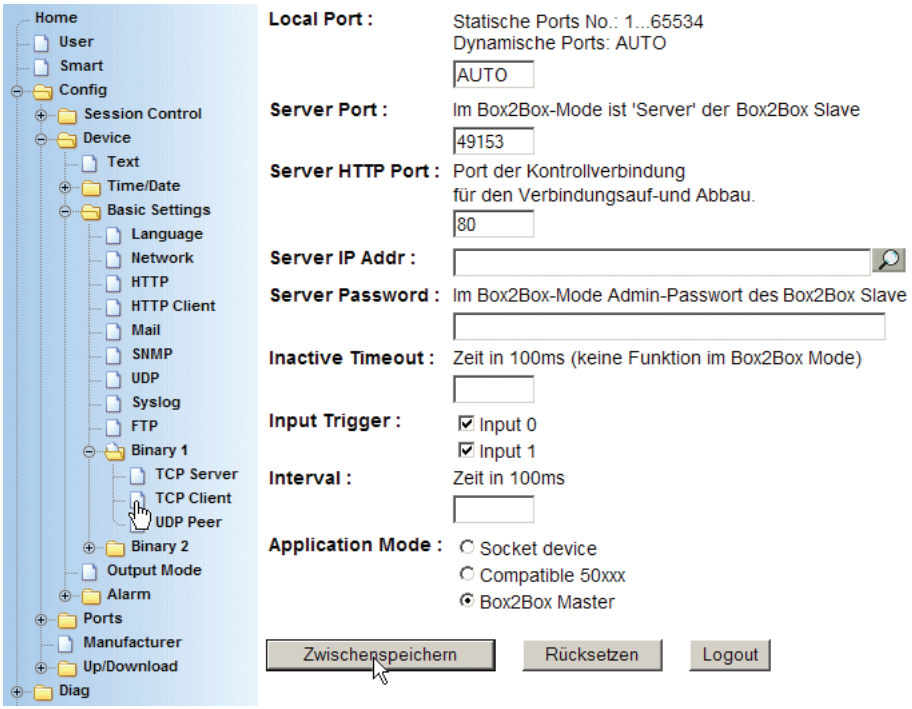
Wählen Sie im Navigationsbaum: *Config >> Device >> Basic Settings >> Binary 1*




Stellen Sie als *Operation Mode* die Betriebsart *TCP-Client* ein und aktivieren Sie *Enable Binary 1*.

Klicken Sie anschließend auf den *Zwischenspeichern*-Button um die Änderungen ans Web-IO zu übertragen.

Wählen Sie nun im Navigationsbaum: *Config >> Device >> Basic Settings >> Binary 1 >> TCP-Client.*




Die folgenden Parameter müssen eingetragen werden:

**Local Port:**
Wenn Ihnen von Ihrem Netzwerkadministrator keine besonderen Vorgaben gemacht werden, kann die werksseitige Einstellung AUTO übernommen werden.

**Server Port:**
Hier muss der beim Slave eingestellte *Local Port* eingetragen werden. Auch hier können die Grundeinstellungen 49153 übernommen werden, wenn vom Administrator keine anderen Vorgaben gemacht werden.

Grund für eine Änderung der werksseitig eingestellten Werte für *Local* und *Slave Port* kann z.B. eine Firewall sein, die nur bestimmte Portzugriffe erlaubt.

**Server HTTP Port**
legt den HTTP Port fest, auf den die Steuerverbindung zur Slave-Box aufgebaut werden soll.

Wenn nicht anders vorgegeben, sollte hier immer Port 80 verwendet werden.

**Server IP Addr:**
Tragen Sie hier die IP-Adresse des Web-IO ein, das als Slave angesprochen werden soll.

**Server Password:**
Hier wird das Administratorpasswort des Slave Web-IO eingetragen. Wurde für den Slave kein Passwort vergeben, bleibt dieses Feld leer.

**Inactive Timeout**
Dieser Parameter hat im Box-to-Box Modus keine Funktion, da eine dauerhafte Verbindung gewünscht ist.

**Input Trigger:**
aktivieren Sie hier die Inputs, die korrespondierenden Outputs beim Slave setzen sollen.

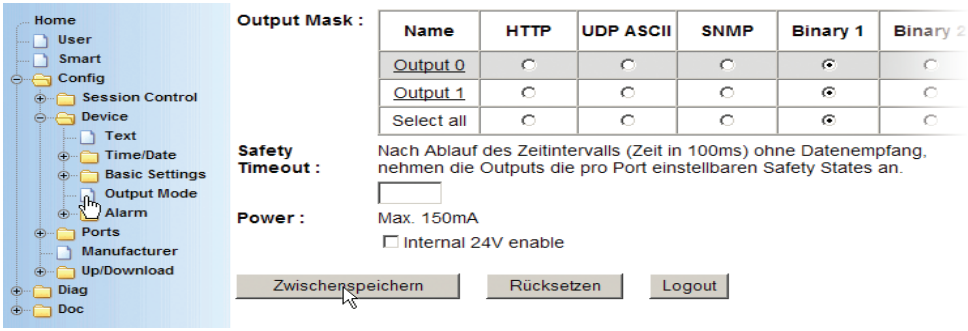
**Interval:**
Ist kein Intervall eingetragen, wird der Zustand der Inputs bei jeder Änderung an die Outputs des jeweils anderen Box-to-Box Partners übergeben. Mit dem Eintrag eines Intervalls, wird der Status auch ohne Änderung zyklisch übergeben.

*Sind zwei Standorte über eine kostenpflichtige ISDN-Strecke miteinander verbunden, sollte auf die Benutzung eines Intervalls verzichtet werden, da die ISDN-Verbindung je nach Timeout und Intervall nie getrennt oder sehr oft neu aufgebaut wird.*

**Application Mode**
Markieren Sie *Box2Box Master*

Nachdem alle Parameter eingegeben wurden, bestätigen Sie durch Klick auf den *Zwischenspeichern*-Button.

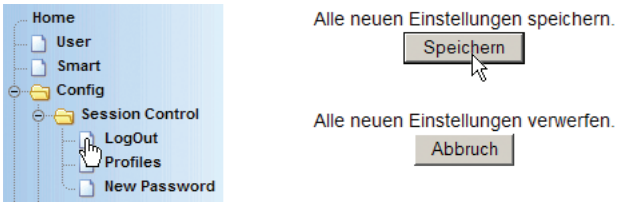
Wählen Sie nun im Navigationsbaum: *Config >> Device >> Output Mode*




Aktivieren Sie hier die Outputs, die von den korrespondierenden Inputs des Slave gesetzt werden sollen für Binary 1 und bestätigen Sie durch Klick auf den *Zwischenspeichern*-Button.

Im Gegensatz zu den Inputs sind die für Box-to-Box aktivierten Outputs für andere Betriebsarten nicht mehr zugänglich.

Nun müssen die neuen Einstellungen noch aktiviert werden. Benutzen Sie den *Logout*-Button oder wählen Sie im Menü *Config >> Session Control >> LogOut.*



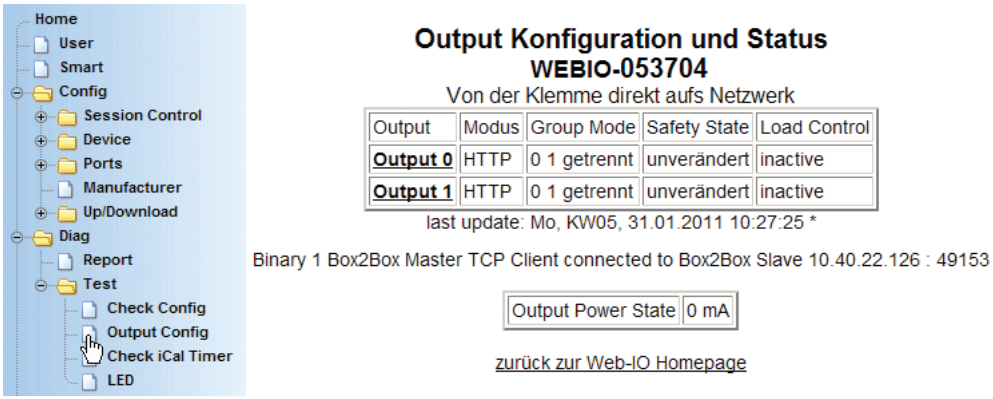

Nach Klick auf den *Speichern*-Button werden alle Einstellungen im Web-IO aktualisiert und die Startseite im Default User Modus neu aufgebaut. Das Master Web-IO versucht darauf hin, sich mit dem Slave Web-IO zu verbinden. Alle hier für Binary 1 beschriebenen Funktionen können natürlich auch unter Binary 2 genutzt werden. So kann z.B ein Web-IO A im Bereich Binary 1 so konfiguriert werden, dass die Inputs/Outputs 0-5 Box to Box mit einem Web-IO B arbeiten. Im Bereich Binary 2 können dann die Inputs/Outputs 6 - 11 so konfiguriert werden, dass sie Box-to-Box mit einem weiteren Web-IO

zusammenarbeiten

### 5.2.3 Box-to-Box Verbindungsstatus ermitteln

Notwendige Zugriffsrechte: *Administrator*

Der Verbindungsstatus einer Box-to-Box Verbindung kann über den Navigationsbaum unter *Diag >> Test >> Output Config* abgerufen werden.




Hier wird angezeigt, in welcher Betriebsart die einzelnen Outputs gerade arbeiten. Zusätzlich wird am Fuß der Webseite der aktuelle Status einer Box-to-Box Verbindung angezeigt.

### 5.2.4 Beenden des Box-to-Box Modus

**Box-to-Box Modus nur beim Master**

Notwendige Zugriffsrechte: *Administrator*

Das Beenden der Betriebsart Box-to-Box muss immer durch entsprechende Konfiguration des Masters erfolgen. Master und Slave Web-IO müssen am Netzwerk angeschlossen sein. Wählen Sie im Navigationsbaum des Masters : *Config >> Device >> Basic Settings >> Binary 1 >> TCP Client* und löschen Sie den Eintrag bei *Server IP Addr*. Setzen Sie ferner den Punkt *Application Mode* auf *Socket Device.*

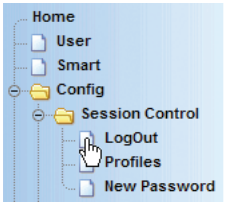
Bestätigen Sie mit Klick auf den *Zwischenspeichern*-Button.

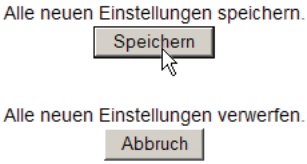
Anschließend setzen Sie unter *Config >> Device >> Basic Settings >> Binary 1 >>* den *Operation Mode* auf *TCP Server .*

Bestätigen Sie mit Klick auf den *Zwischenspeichern*-Button.

Nun wählen Sie im Navigationsbaum des Masters : *Config >> Device >> Output Mode* und setzen die Outputs, die nicht mehr Box-to-Box arbeiten sollen, auf HTTP.

Bestätigen Sie mit Klick auf den *Zwischenspeichern*-Button. Nun müssen die geänderten Einstellungen noch aktiviert werden. Benutzen Sie den *Logout*-Button oder wählen Sie im Menü *Config >> Session Control >> LogOut.*




Nach Klick auf den *Speichern-Button* werden alle Einstellungen im Web-IO aktualisiert und die Startseite im Default User Modus neu aufgebaut.

**Box-to-Box Modus beim Slave Web-IO beenden**

Notwendige Zugriffsrechte: *Administrator*

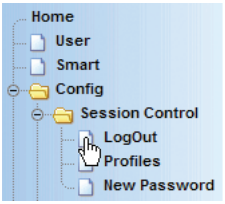
Wählen Sie im Navigationsbaum des Slave: *Config >> Device >> Basic Settings >> Binary 1 >> TCP Server* und setzen Sie den Punkt *Application Mode* auf *Socket Device.*

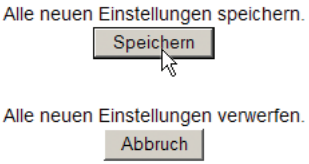
Bestätigen Sie mit Klick auf den *Zwischenspeichern*-Button.

Nun wählen Sie im Navigationsbaum *Config >> Device >> Output Mode* und setzen die Outputs, die nicht mehr Box-to-Box arbeiten sollen, auf HTTP.

Bestätigen Sie mit Klick auf das *Zwischenspeichern*-Button.

Nun müssen die geänderten Einstellungen noch aktiviert werden. Benutzen Sie den *Logout*-Button oder wählen Sie im Menü *Config >> Session Control >> LogOut.*




Nach Klick auf den *Speichern-Button* werden alle Einstellungen im Web-IO aktualisiert und die Startseite im Default User Modus neu aufgebaut.

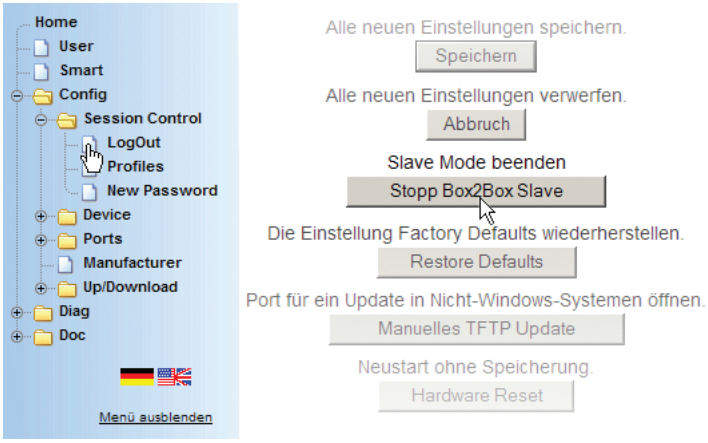
### 5.2.5 Box-to-Box Slave - Bindung an den Master aufheben

Um z.B. nach Unterbrechung der Netzwerkinfrastruktur Verbindung mit dem Master aufzunehmen, speichert das Slave Web-IO die IP-Adresse des Masters (für den Anwender nicht sichtbar).

Soll das Slave Web-IO mit einem neuen Master Web-IO zusammenarbeiten oder hat sich die IP-Adresse des Masters geändert, muss dieser unsichtbare Adresseintrag gelöscht werden.

Wählen Sie dazu im Navigationsbaum *Config >> Session Control >> LogOut*.

Im Konfigurationsframe sollte ein zusätzlicher Button *Stopp Box2Box Slave* vorhanden sein.




Sollte dieser Button nicht angezeigt werden, Klicken Sie zunächst den *Reset*-Button. Das Web-IO wird neu gestartet.

Nach erneutem Login und Aufruf von *Config >> Session Control >> LogOut* ist das *Stopp Box2Box Slave-Button* vorhanden. Durch Klick wird der Eintrag im Slave zurückgesetzt.

### 5.2.6 Box-to-Box Hinweise und Grenzen

**Laufzeiten**
Bitte beachten Sie, dass bei Datenübertragung über ein Netzwerkprotokoll systembedingt Laufzeitverzögerungen auftreten.

Im Gegensatz zu einer direkten Kabelanbindung, bei der ein Schaltsignal direkt am Aktor ansteht, müssen bei Verbindung über Web-IO im lokalen Netzwerk Latenzzeiten zwischen 20 und 50 ms einkalkuliert werden.

Sind aktive Komponenten wie Router, Bridges oder gar ISDN-Strecken in den Übertragungsweg einbezogen, erhöht sich die Latenzzeit entsprechend, so dass es zu Laufzeiten von 500ms und höher kommen kann.

**Box-to-Box zu mehreren Partnern**
Es ist nicht möglich, Signale von einem Input an den Outputs mehrerer Web-IOs auszugeben. Es besteht aber die Möglichkeit, über die Modi Binary 1 und Binary 2 von einem Master Web-IO z.B die Inputs 0 - 5 an Slave A zu binden, aber die Inputs 6 - 11an Slave B zu knüpfen.

**Mischen der verschiedenen Web-IO Modelle**
Es besteht grundsätzlich die Möglichkeit, verschiedene Web-IO Modelle Box-to-Box miteinander zu verbinden. Als Bedingung gilt in diesem Fall, dass das Web-IO mit der geringeren Anzahl an Ports als Master konfiguriert wird.

**1. Beispiel:**
Ein Web-IO 2x Digital Input, 2x Digital Output soll mit einem Web-IO 12x Digital Input, 12x Digital Output Box-to-Box verbunden werden.

Das Web-IO 12x Digital Input, 12x Digital Output wird in diesem Fall unter Binary 1 als Box-to-Box Slave konfiguriert, wobei die Input Trigger für Input0 und Input1 gesetzt werden. Ferner werden im Bereich Output Mode die Outputs 0 und 1 für Binary 1 freigeschaltet.

Anschließend wird die Konfiguration abgespeichert.

Das Web-IO 2x Digital Input, 2x Digital Output wird ganz normal als Box-to-Box Master für Binary 1 konfiguriert.

**2. Beispiel**
Es sollen zwei Web-IO 2x Digital Input, 2x Digital Output mit einem Web-IO 12x Digital Input, 12x Digital Output im Box-to-Box Modus verbunden werden.

Wie im ersten Beispiel, wird zunächst das Web-IO 12xDigital als Slave konfiguriert. Das geschieht sowohl für Binary 1 als auch für Binary 2. Für Binary 1 werden die Input Trigger 0 und 1 gesetzt, für Binary 2 die Input Trigger 2 und 3. Darüber hinaus werden im Bereich Output Mode die Outputs 0 und 1 für Binary 1 und die Outputs 2 und 3 für Binary 2 konfiguriert.

Anschließend wird die Konfiguration abgespeichert.

Das 1. Web-IO 2x Digital Input, 2x Digital Output wird ganz normal als Box-to-Box Master für Binary 1 konfiguriert. Beim 2. Web-IO 2x Digital Input, 2x Digital Output wird in der gleichen Weise für Binary 2 verfahren.

Nachdem die Einstellungen bei allen beteiligten Web-IO gespeichert wurden, folgt Output 0 des Web-IO 12x Digital Input, 12x Digital Output Änderungen an Input 0 vom 1. Web-IO 2x Digital Input, 2x Digital Output. Ändert sich Input 1 am 2. Web-IO 2x Digital Input, 2x Digital Output zieht das eine Änderung an Output 2 des Web-IO 12x Digital Input, 12x Digital Output nach sich.

**Alte und neue Geräte als Box-to-Box Kombination**
Es ist nicht möglich, ältere E/A-Com-Server 50xxx mit einem Web-IO Box-to-Box zu betreiben. Web-IO, die im Box-to-Box Modus zusammenarbeiten, sollten den gleichen Firmwarestand haben

**Probleme bei Box-to-Box**
Vor allem wenn sich die IP-Adressen eines Box-to-Box Paares z.b. durch Umstellung der Infrastruktur geändert hat, kommt es vor, das der Master keine Verbindung mehr zum Slave aufbauen kann oder umgekehrt.

Unter *Config >> Session Control >> LogOut* sollte in solchen Fällen beim Slave ein zusätzliches Button *Stopp Box2Box Slave* vorhanden sein.

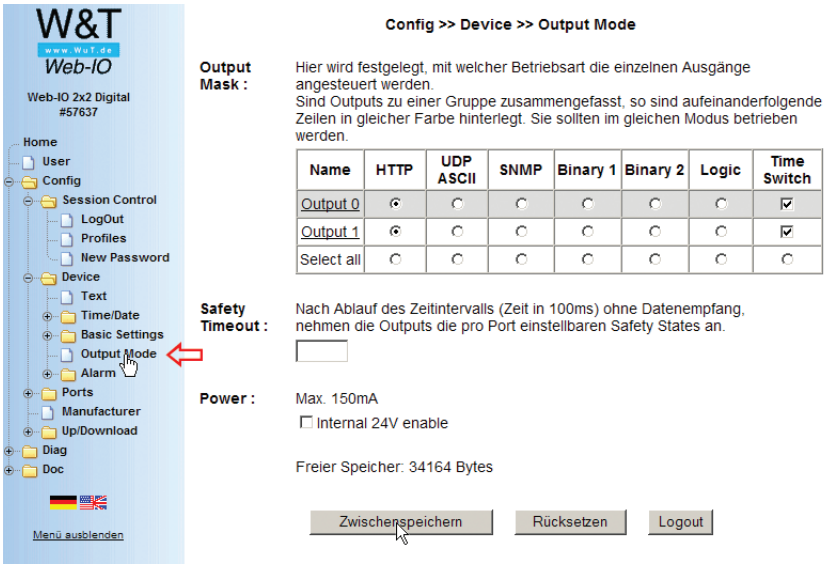
Ein Klick auf diesen Button setzt ggf. vorhandene alte Verbindungsparameter zurück und öffnet die Slave-Box so für eine neue Verbindung.

## 5.3 Time Switch - Web-IO als Schaltuhr

Das Web-IO Modell 57637 bietet über den normalen Funktionsumfang hinaus die Möglichkeit, die Outputs zeitgesteuert zu schalten.

### 5.3.1 Aktivieren er Outputs für Time Switch

Wählen Sie im Navigationsbaum des Web-IO *Config >> Device >> Output Mode*




Aktivieren Sie hier die Outputs, die zeitgesteuert geschaltet werden sollen und bestätigen Sie durch Klick auf den *Zwischenspeichern*-Button.

Wenn das Web-IO im *Output Mode Time Switch* arbeitet, ist es nach wie vor über die eingestellte Grundbetriebsart schaltbar.

Das bedeutet, dass der durch die Schaltuhr eingestellte Output-Zustand z.B. durch Bedienung im Browser geändert werden kann. Das nächste zeitgesteuerte Schalten ändert aber den manuell eingestellten Zustand wieder.

*Der parallele Zugriff von Logic und Time Switch ist nicht erlaubt.*

Sichern Sie die Einstellungen durch *Logout* und *Speichern* bevor Sie mit der weiteren Konfiguration fortfahren.

### 5.3.2 Schaltvorgaben laden

Das Web-IO hat keine Konfigurationsoberfläche zum Einstellen der Schaltuhr. Stattdessen gibt es zwei Möglichkeiten, die Steuerdaten in Form einer iCal-Kalenderdatei ins Web-IO zu laden:

- iCal-Datei direkt übers Internet aus einem Online-Kalender laden
- iCal-Datei vom lokalen PC laden

### 5.3.3 Schaltzeiten aus Online-Kalender laden

*Konfigurieren Sie zunächst die Netzwerkeinstellungen (IP-Adresse, Subnet-Mask, Gateway und DNS-Server) des Web-IO und übernehmen Sie die Einstellungen durch* Logout *und* Speichern*.*

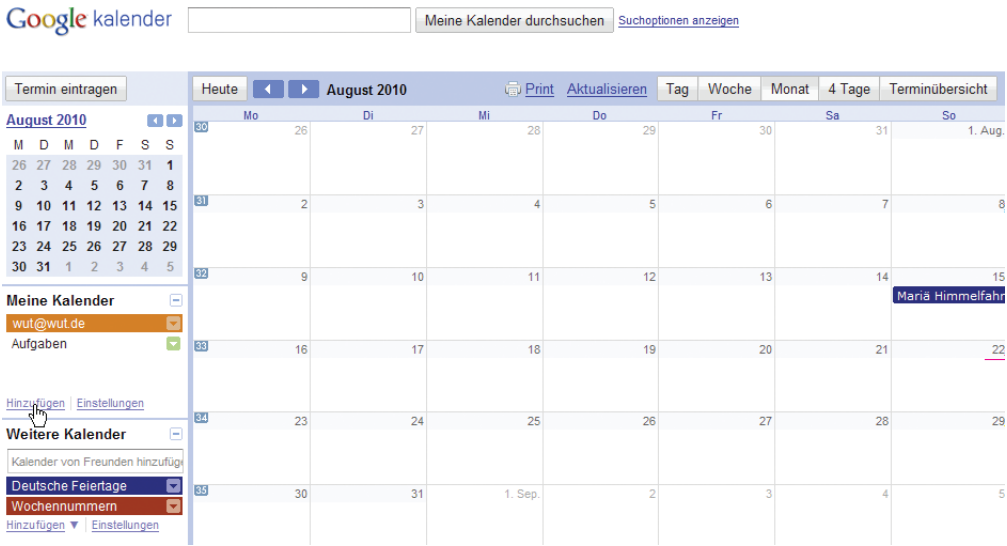
Als Beispiel für das Verwalten der Schaltzeiten zeigen wir hier den Umgang mit dem Kalender des Internet-Suchanbieters Google. Um die Funktionalität des Google-Kalenders nutzen zu können, muss unter www.google.de ein entsprechendes Benutzerkonto angelegt werden.

*Bitte beachten Sie die Nutzungsbedingungen des Google-Kontos, denen Sie mit Einrichten des Kontos zustimmen.*

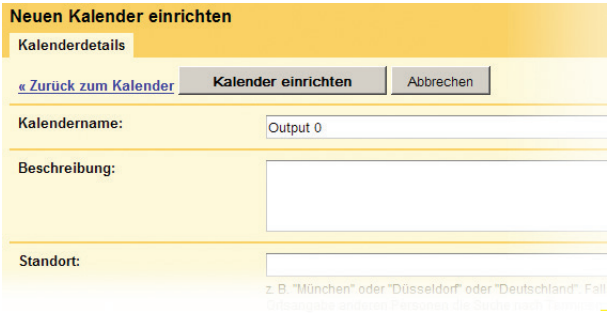
Es kann natürlich auch jeder andere Online-Kalender genutzt werden, der einen öffentlichen Link auf die dem Kalender entsprechende iCal-Datei bereit stellt.

Jeder der beiden Outputs benötigt einen eigenen Kalender,

in dem die Schaltvorgaben eingetragen werden. Bei Google können unter einem Kundenkonto mehrere Kalender gepflegt werden.




Um einen neuen Kalender anzulegen, wählen Sie unter *Meine Kalender* den Link *Hinzufügen*.

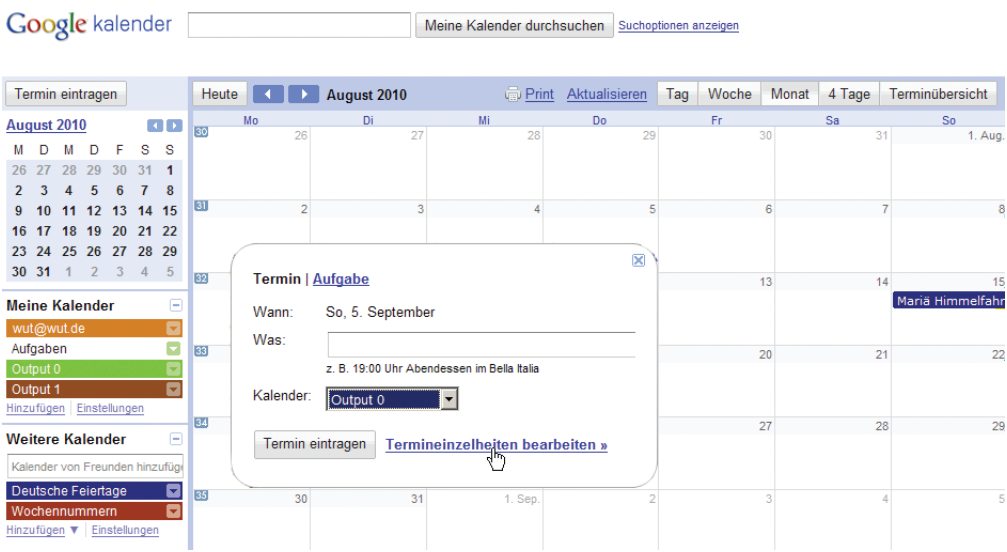



Es spielt keine Rolle, welchen Namen Sie dem Kalender geben. Wir empfehlen aber, sinngebende Namen zu verwenden, um bei Einsatz mehrerer Web-IOs den Überblick zu behalten.

Weitere Einträge oder Änderungen sind an dieser Stelle nicht nötig. Um den Kalender anzulegen, klicken Sie auf *Kalender einrichten.*

Wenn für beide Outputs ein Kalender eingerichtet wurde, können die gewünschten Schaltzeiten eingetragen werden.

Der Google-Kalender lässt sowohl einmalige als auch wiederholende Termine zu. Um einen neuen Termin anzulegen, klicken Sie auf *Termin eintragen* oder auf den gewünschten Tag im Kalender.



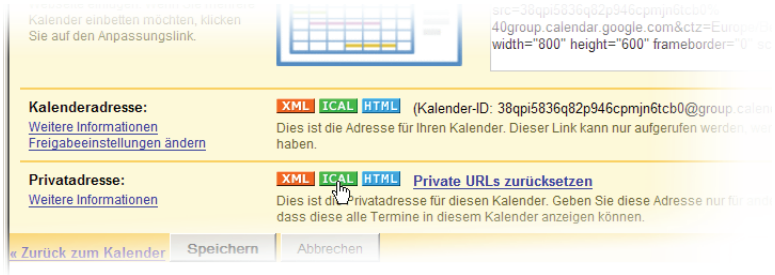

Wählen Sie al *Kalender* den Kalender für den entsprechenden Output aus und klicken Sie auf *Termineinzelheiten bearbeiten*.

Bestimmen Sie alle Termineinzelheiten und klicken dann auf *Speichern*. Es spielt keine Rolle, welchen Namen Sie unter *Was* eintragen. Das Web-IO schaltet zum Beginn jedes gefundenen Termins den Output ein und mit Ende des Termins wieder aus.

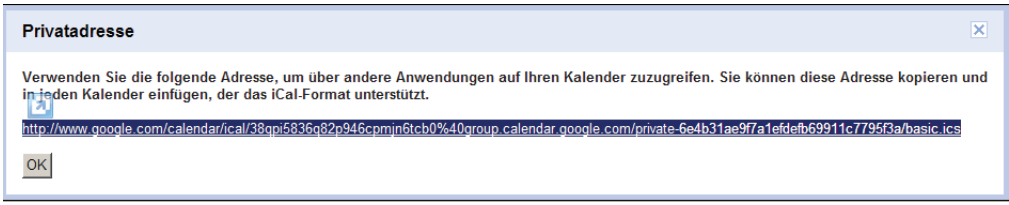
Konfigurieren Sie auf diese Weise all gewünschten Schaltzeiten.

Abschließend klicken Sie unter *Meine Kalender* auf den Link *Einstellungen*.

Wählen Sie den Kalender für Output 0 aus.

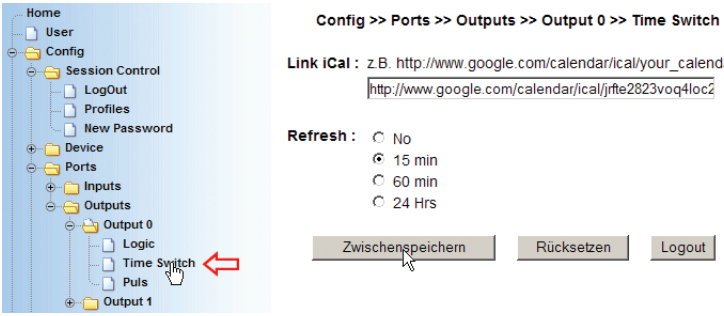



Im unteren Bereich der nun folgenden Webseite finden Sie den Punkt *Privatadresse.* Klicken Sie hier auf *iCal*.




Daraufhin wird der Online-Link auf die zum Kalender gehörende iCal-Datei angezeigt. Markieren Sie diesen Link und kopieren Sie ihn mittels *<Strg> - C* in die Zwischenablage.

Wählen Sie nun im Navigationsbaum des Web-IO *Config >> Ports >> Outputs >> Output 0 >> Time Switch.*



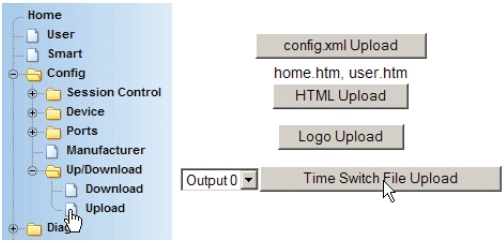

**Link iCal**

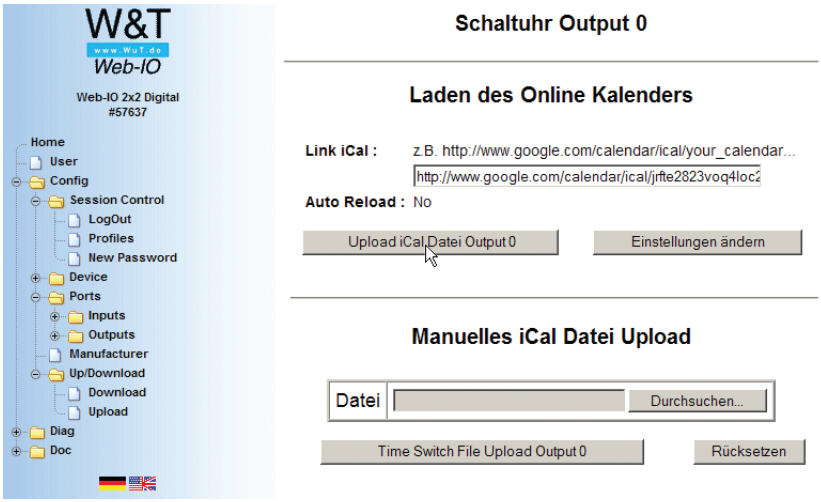
Fügen Sie hier den in die Zwischenablage kopierten Link mit *<Strg> - C* ein.

**Refresh**

An dieser Stelle wird festgelegt, in welchem Zyklus das Web-IO die Kalenderdaten vom Online-Kalender herunterlädt. Wird *No,* also kein automatischer Abgleich gewählt, muss die Datei von Hand geladen werden.

Wählen Sie dazu im Navigationsbaum des Web-IO *Config >> Up/Download >> Upload* und klicken Sie auf *Time Switch File Upload Output 0.*






Klicken Sie auf *Upload iCal Datei Output 0*.

In gleicher Weise muss für Output 1 verfahren werden.

Nun müssen die neuen Einstellungen noch aktiviert werden. Benutzen Sie den *Logout*-Button oder wählen Sie im Menü *Config >> Session Control >> LogOut.*

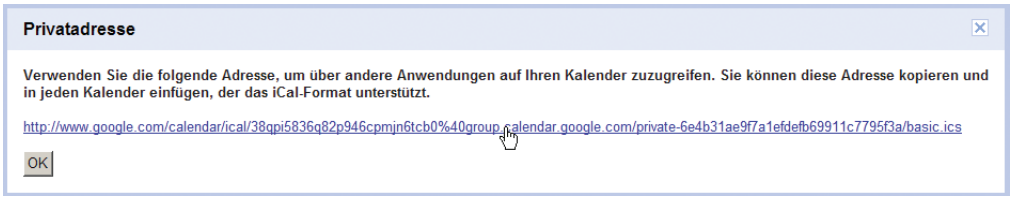
### 5.3.4 Schaltzeiten als iCal-Datei laden

Sollte das Web-IO keinen Zugang zum Internet haben, bietet sich als zweite Alternative die Möglichkeit, die Schaltzeiten in Form einer iCal-Datei von der Festplatte eines lokalen PC ins Gerät zu laden.

Auch in diesem Fall können die Schaltzeiten über einen Online-Kalender erstellt werden. Die Vorgehensweise ist identisch mit der vorangehend beschriebenen.

Allerdings muss die iCal-Datei zunächst auf einen mit dem Internet verbundenen PC heruntergeladen werden.

Beim Google-Kalender müssen Sie dazu nur auf den Link der iCal-Datei klicken.




Die heruntergeladene iCal-Datei kann dann mittels USB-Stick auf einen PC übertragen werden, der Verbindung zum Web-IO hat.

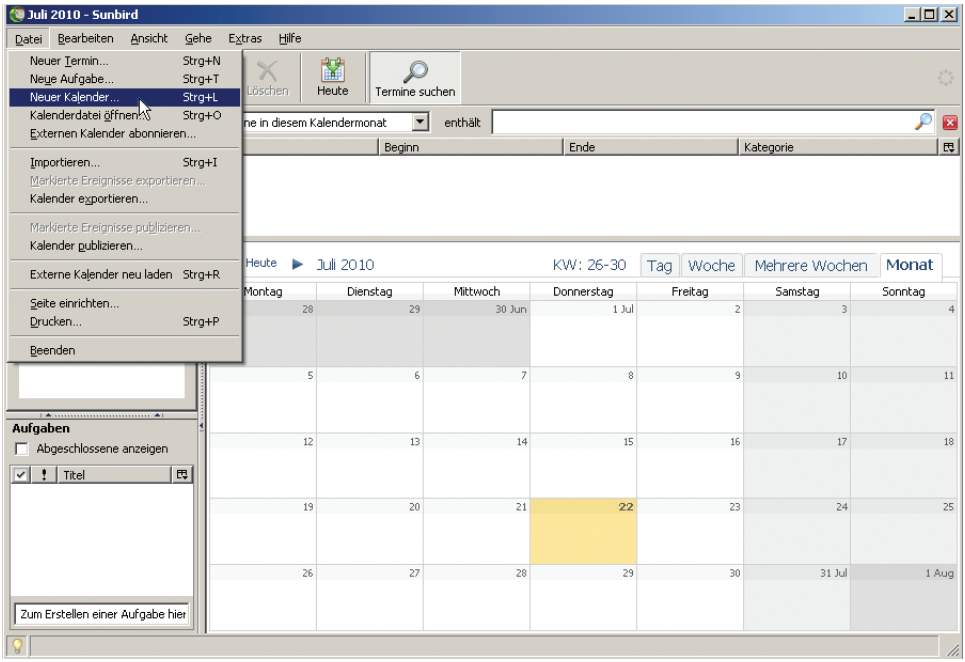
Natürlich kann die iCal-Datei auch mit jedem Kalenderprogramm erstellt werden, das über eine iCal-Exportfunktion verfügt.

Als Freeware-Variante bietet sich z.B. Sunbird von Mozilla an.

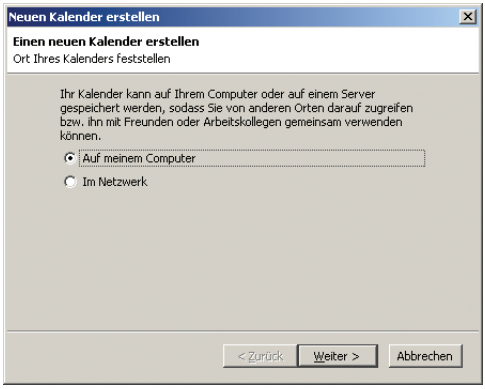
Kostenloser Download unter:

http://www.sunbird-kalender.de/downloads.php

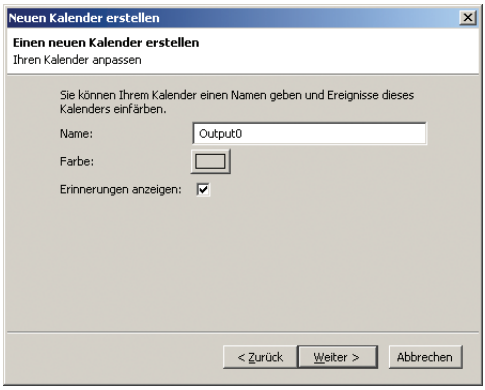
Die Installation von Sunbird ist selbsterklärend.




Nach dem ersten Start wählen Sie im Menü *Datei >> Neuer Kalender.*

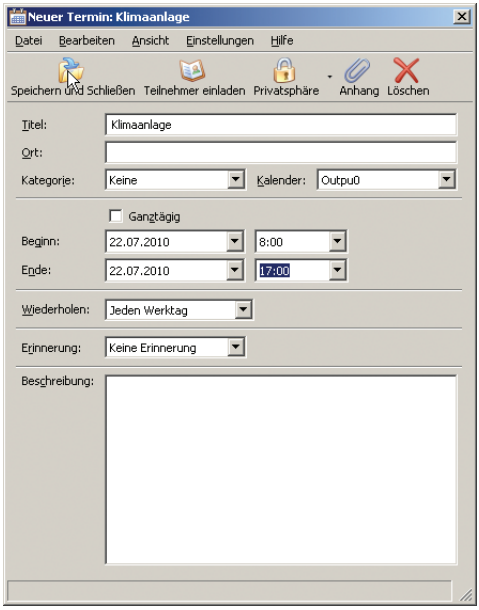



Der Kalender sollte auf dem lokalen PC gespeichert werden.




Geben Sie dem Kalender einen sinngebenden Namen und bestätigen Sie mit *Weiter*.

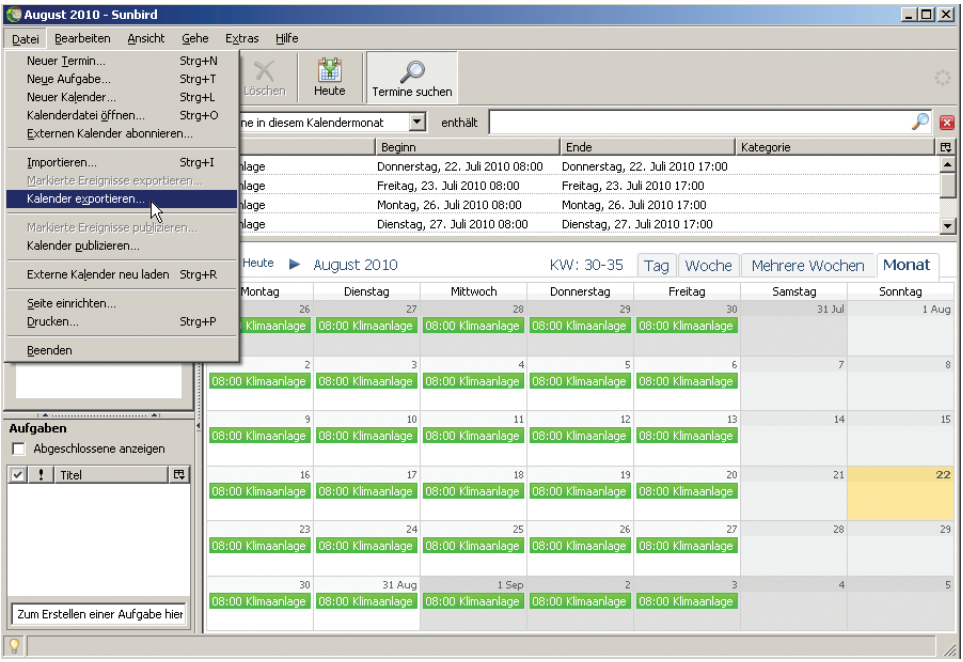
Sunbird lässt sowohl einmalige als auch wiederholende Termine zu. Um einen neuen Termin anzulegen, doppelklicken Sie auf den gewünschten Tag im Kalender.




Bestimmen Sie alle Termineinzelheiten und klicken dann auf *Speichern und Schließen* . Es spielt keine Rolle, welchen Namen Sie unter *Titel* eintragen. Das Web-IO schaltet zum Be-

ginn jedes gefundenen Termins den Output ein und mit Ende des Termins wieder aus.

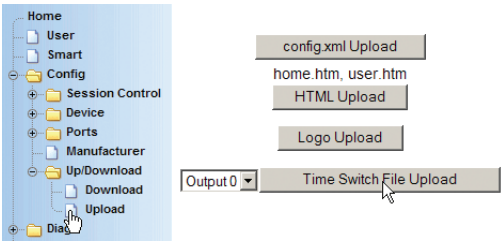
Konfigurieren Sie auf diese Weise alle gewünschten Schaltzeiten.



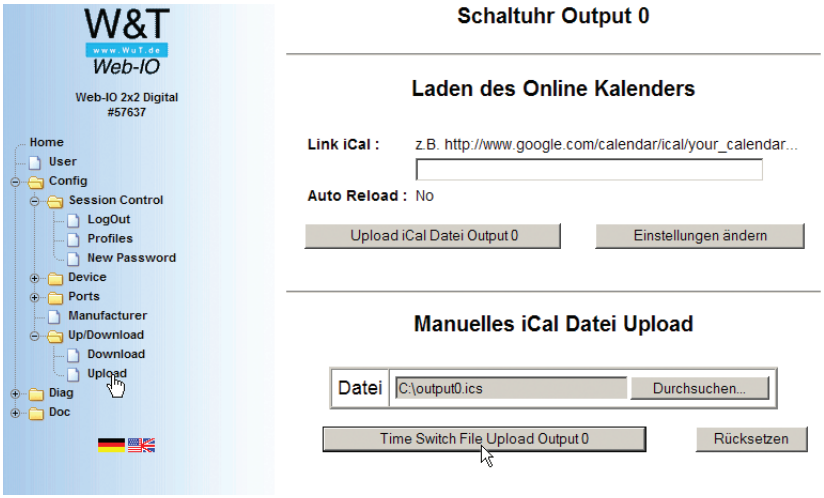

Abschließend klicken Sie unter *Datei* auf *Kalender exportieren*.

Speichern Sie die iCal-Datei auf der Festplatte des PC.

Wählen Sie nun im Navigationsbaum des Web-IO *Config >> Up/Download >> Upload* und klicken Sie auf *Time Switch File Upload Output 0.*




Um die iCal-Datei ins Web-IO zu laden, wählen Sie die gewünschte Datei aus und klicken auf *Time Switch File Upload Output 0*.




In gleicher Weise muss für Output 1 verfahren werden.

Nun müssen die neuen Einstellungen noch aktiviert werden. Benutzen Sie den *Logout*-Button oder wählen Sie im Menü *Config >> Session Control >> LogOut*.

### 5.3.5 Schaltuhr Vorschau

Nach Abschluss der Schaltuhrkonfiguration beginnt das Web-IO mit der Abarbeitung des eingespielten Zeitsteuerprogramms.

**Vorschau iCal Timer**

Schaltuhr Vorschau (Time Switch)

| Datum | Uhrzeit | Output 0 | Output 1 |
|---|---|---|---|
| 22.07.2010 | 13:01:38 | ON | OFF |
| 22.07.2010 | 17:00:00 | OFF | OFF |
| 22.07.2010 | 21:00:00 | OFF | ON |
| 23.07.2010 | 06:00:00 | OFF | OFF |
| 23.07.2010 | 08:00:00 | ON | OFF |

Eine Vorschau der Schaltereignisse der nächsten 24 Stunden kann von der Home-Seite des Web-IO abgerufen werden. Klicken Sie dazu einfach auf den Link *Schaltuhr Vorschau (Time Switch).*

*Für eine langfristige, detaillierte Vorschau können die iCal-Dateien unter Config >> Up/Download >> Download > iCal Datei heruntergeladen und in einem Kalenderprogramm geöffnet werden.*

### 5.3.6 Besonderheiten der Web-IO Schaltuhr

Es ist möglich, sowohl den Abgleich mit einem Online-Kalender zu konfigurieren als auch eine iCal-Datei hochzuladen.

In der Praxis ist das nur sinnvoll, wenn vorübergehend abweichend vom normalen Betrieb geschaltet werden soll.

Wenn für den Online-Abgleich ein Zyklus konfiguriert wurde, gelten die manuell hochgeladenen Schaltzeiten nur solange, bis der nächste zyklusgemäße Online-Abgleich stattfindet. Danach gelten wieder die Schaltzeiten des Online-Kalenders.

Unabhängig davon, auf welchem Weg das Web-IO die Zeitschaltvorgaben bekommen hat, wenn unter *Config, Device >> Outputmode* zusätzlich zu Time Switch eine der Grundbe-

triebsarten

- HTTP
- UDP ASCII
- SNMP
- Binary 1
- Binary 2

aktiviert ist, kann der aktuelle Schaltzustand jederzeit über die eingestellte Zugriffsart geändert werden. Der nächste zeitgesteuerte Schaltvorgang würde allerdings wieder den Output-Zustand bestimmen.

Darüber hinaus prüft das Web-IO beim zyklischen Nachladen der Kalenderdatei, welcher Zustand vom Kalender z.Zt. vorgegeben ist und setzt den Output entsprechend.

*Der maximale interne Speicher für jede iCal-Datei ist 32KB groß. Vor allem bei sich dynamisch ändernden Kalendern, sollten bereits vergangene Termine gelöscht werden, damit die Datei diese Größe nicht überschreitet.*

## 5.4 Integration in bestehende Systeme

Damit das Web-IO problemlos auch in bestehende Systeme Integriert werden kann, unterstützt es einige in der Automatisierung und Netzwerktechnik übliche Standardkommunikationswege:

- SNMP
- OPC
- Modbus

### 5.4.1 SNMP - In- und Outputs in Managementsystemen

Das Web-IO Digital kann über das SNMP-Protokoll in bestehende Netzwerkmanagementsysteme eingebunden werden.

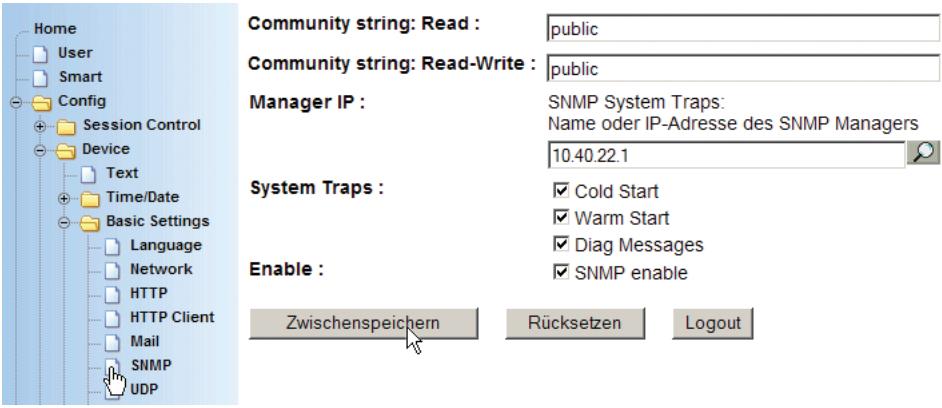
Dabei kann sowohl die gesamte Konfiguration, als auch das Steuern und Überwachen von Inputs, Countern und Outputs über SNMP erfolgen.

*Die Private-MIB steht unter http://www.wut.de zum Download zur Verfügung. Navigieren Sie von der Datenblattseite des entsprechenden Web-IO in den Bereich Tools.*

**Freischalten des SNMP-Zugriffs**

Notwendige Zugriffsrechte: *Administrator*

Wählen Sie im Navigationsbaum des Web-IO *Config >> Device >> Basic Settings >> SNMP*




**Community String: Read**
Tragen Sie hier den Community String für SNMP Zugriff mit Leseberechtigung ein.

**Community String: Read-Write**
Tragen Sie hier den Community String für SNMP Zugriff mit Lese- und Schreibberechtigung ein.

**Manager IP:**
IP-Adresse des SNMP Management-Systems

**System Traps:**
Das Web-IO bietet die Möglichkeit, bei Kalt- oder Warmstart (Hardware-Reset und Power-ON oder Software-Reset) einen System Trap an den eingetragen SNMP Manager abzusetzen.

Markieren Sie entsprechend *Cold Start* oder *Warm Start*.

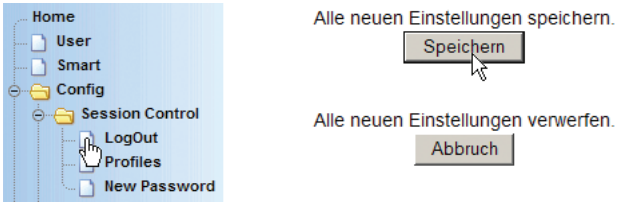
**Diag Messages:**
Mit dieser Option lassen sich Fehler- und Diagnosemeldungen als SNMP-Trap an das SNMP-Managementsystem senden. Solange der Web-IO interne Fehlerspeicher nicht gelöscht wird, wird ein Fehler nur beim ersten Auftreten als Trap gesendet. Welche Fehler aktuell anliegen, kann über die Diag-Seite im Browser abgerufen werden.

siehe: http://<IP-Adresse des Web-IO/diag

**Enable:**
An dieser Stelle muss SNMP Enable aktiviert werden, um mit dem Web-IO auf SNMP-Ebene arbeiten zu können.

Bestätigen Sie Ihre Eingaben durch Klick auf den *Zwischen-speichern*-Button..

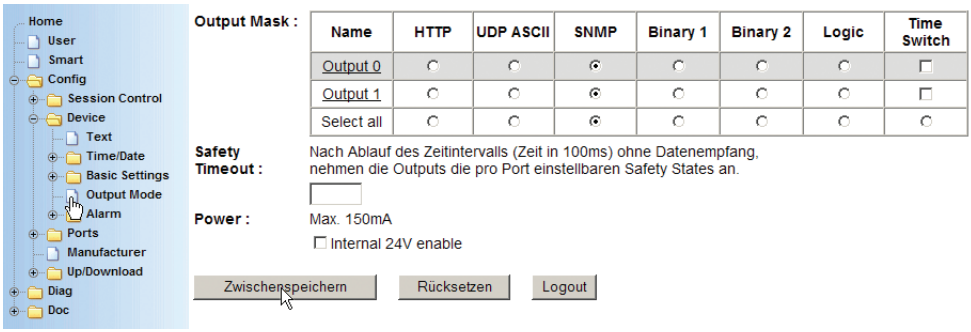
Wenn keine weiteren Konfigurationsänderungen mehr erfolgen sollen, müssen die neuen Einstellungen noch aktiviert werden. Benutzen Sie den *Logout*-Button oder wählen Sie im Menü *Config >> Session Control >> LogOut.*




Nach Klick auf den *Speichern*-Button werden alle Einstellungen im Web-IO aktualisiert und die Startseite im Default User Modus neu aufgebaut.

**Aktivieren der Outputs für SNMP**
Notwendige Zugriffsrechte: *Administrator*

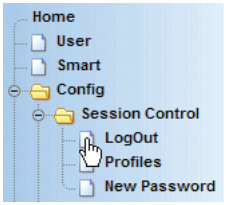
Wählen Sie im Navigationsbaum des Web-IO *Config >> Device >> Output Mode*

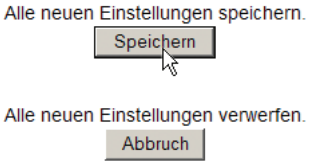



Aktivieren Sie hier die Outputs, die über das SNMP-Protokoll gesteuert werden sollen und bestätigen Sie durch Klick auf

den *Zwischenspeichern*-Button.

Nun müssen die neuen Einstellungen noch aktiviert werden. Benutzen Sie den *Logout*-Button oder wählen Sie im Menü *Config >> Session Control >> LogOut*.




Nach Klick auf den *Speichern*-Button werden alle Einstellungen im Web-IO aktualisiert und die Startseite im Default User Modus neu aufgebaut.

**Die wichtigsten Schritte bei SNMP-Zugriff**

Die von W&T bereitgestellte Private MIB kann für alle Web-IO Digital Geräte eingesetzt werden. Für die verschiedenen Modelle sind getrennte OID-Zweige vorhanden.

Die hier vorgestellte Vorgehensweise ist beispielhaft für das Web-IO 12xDigital Input, 12xDigital Output ausgelegt.

Sobald der SNMP-Zugriff wie eingangs beschrieben freigeschaltet wurde, ist jederzeit ein lesender Zugriff möglich.

Um schreibenden Zugriff auf das Web-IO zu bekommen, ist es zunächst nötig, eine SNMP-Session aufzubauen. Das gilt vor allem dann, wenn die Outputs per SNMP gesetzt werden sollen.

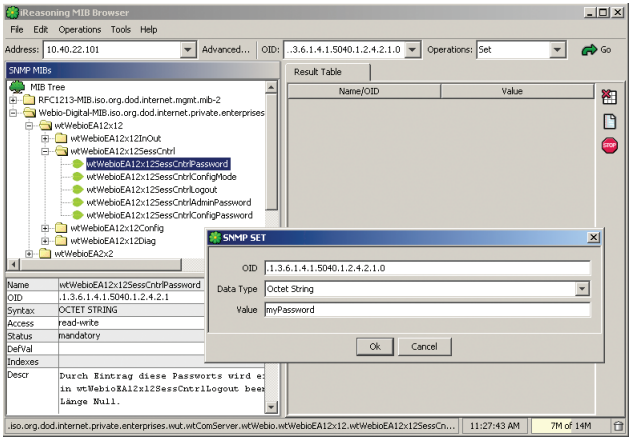
**Session Login**

Um eine Session zu öffnen, muss das Administrator-Passwort in die Variable

```
wtWebioEA12x12SessCntrPassword
```

eingetragen werden.

*Eine geöffnete Session bleibt für 5 Minuten erhalten und wird danach automatisch verworfen.*



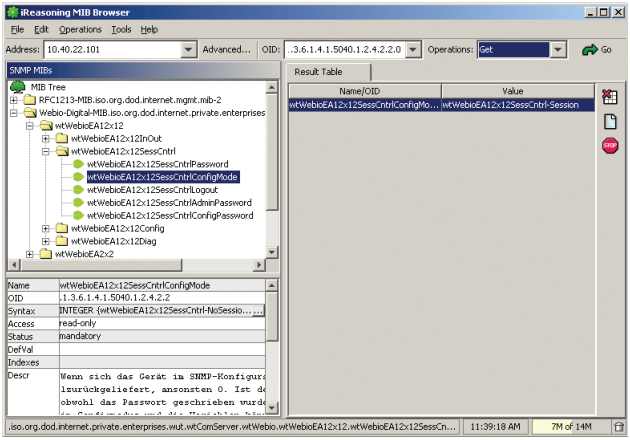

**Session auf Gültigkeit prüfen**

Durch Auslesen der Variablen

`wtWebioEA12x12SessCntrlConfigMode`

kann überprüft werden, ob die Session erfolgreich geöffnet wurde.

1 = Session geöffnet, Web-IO i Konfigurationsmodus
0 = Öffnen der Session ist fehlgeschlagen. Überprüfen Sie, ob ggf. das Passwort falsch angegeben wurde.



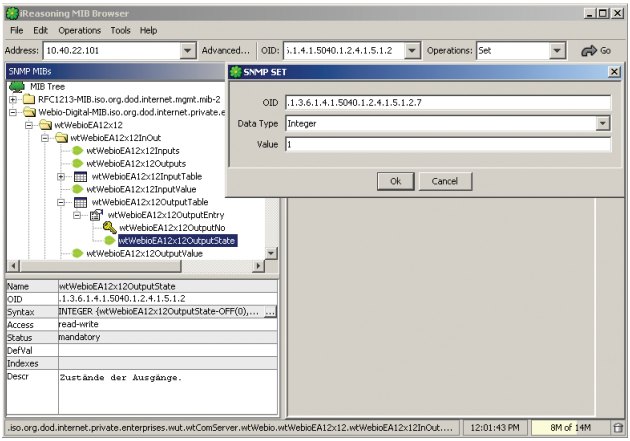

Nach erfolgreichem Öffnen der Session, können über die in der Private-MIB definierten Variablen beliebige Konfigurationsänderungen bzw. Input und Output-Zugriffe erfolgen.

### Einen Output verändern

Um nach einem erfolgreichen Session-Login einzelne Outputs in ihrem Zustand zu verändern, wird die SNMP-Variable

`wtWebioEA12x12OutputState`

benutzt. Um zu bestimmen, welcher Output verändert werden soll, muss an die entsprechende OID, getrennt durch einen Punkt, die Nummer des Outputs + 1 angehängt werden.



Als Value wird 1 für ON und 0 für OFF übergeben. Hier wird z.B. Output 6 auf ON gesetzt.

### Alle Outputs gleichzeitig verändern

Neben dem Setzen einzelner Outputs, besteht die Möglichkeit, alle Outputs auf einmal zu verändern. Dazu wird die SNMP-Variable

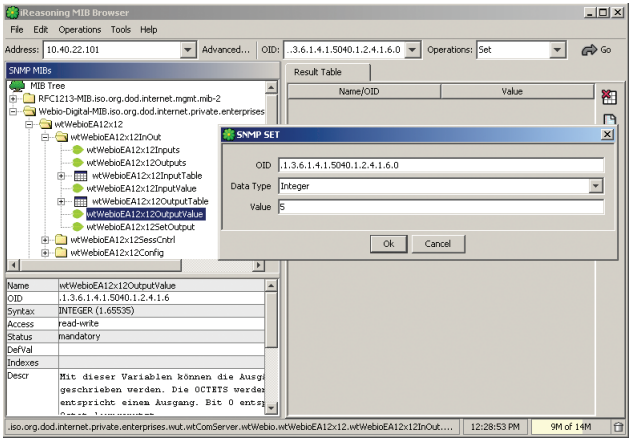
`wtWebioEA12x12OutputValue`

benutzt. Übergeben wird der Integerwert, der dem binären Muster der Outputs entspricht.

Beispiel: Output 0 und Output 2 sollen auf ON gesetzt werden. Alle anderen Outputs sollen den Status OFF annehmen.

Der zu übergebende Wert wird berechnet wie folgt:

$1x2^0 + 0x2^1 + 1x2^2 + 0x2^3 + 0x2^4 + 0x2^5 + 0x2^6 + 0x2^7 + 0x2^8 + 0x2^9 + 0x2^{10} + 0x2^{11} = 5$




### Mehrere Outputs verändern

Als letzte Variante bietet das Web-IO die Möglichkeit, mehrere Outputs zeitgleich zu setzen, andere hingegen in ihrem alten Zustand zu belassen.

Hierzu wird mit einer Output-Maske und dem Output-Wert gearbeitet.

Mit der Output-Maske wird festgelegt, welche Outputs verändert werden sollen. Der Output-Wert hingegen legt fest welchen Zustand diese Outputs annehmen sollen.

Beispiel:
Output 1 soll auf ON geschaltet werden,
Output 2 soll auf OFF geschaltet werden,
Output 4 soll auf ON geschaltet werden,
Output 5 soll auf OFF geschaltet werden,
Output 10 soll auf ON geschaltet werden,
Output 11 soll auf ON geschaltet werden,

Verändert werden sollen also die Outputs:

1,2,4,5,10 und 11.

Die Output-Maske berechnet sich:

$0x2^0 + 1x2^1 + 1x2^2 + 0x2^3 + 1x2^4 + 1x2^5 + 0x2^6 + 0x2^7 + 0x2^8 + 0x2^9 + 1x2^{10} + 1x2^{11} = 3126$

In hexadezimaler Schreibweise entspricht das 0xC36.

Um den Output-Wert zu berechnen müssen nur die Bitwertigkeiten addiert werden, die dem Zustand ON - also 1 - entsprechen.

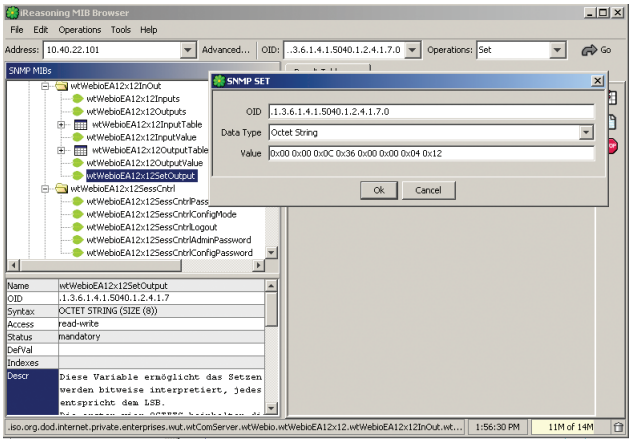
$1x2^1 + 1x2^4 + 1x2^{10} = 1042$

In hexadezimaler Schreibweise entspricht das 0x412.

Output-Maske und -Wert werden über die SNMP-Variable

`wtWebioEA12x12SetOutput`

als 8 Byte Oktetstring übergeben.




**Konfiguration ändern**

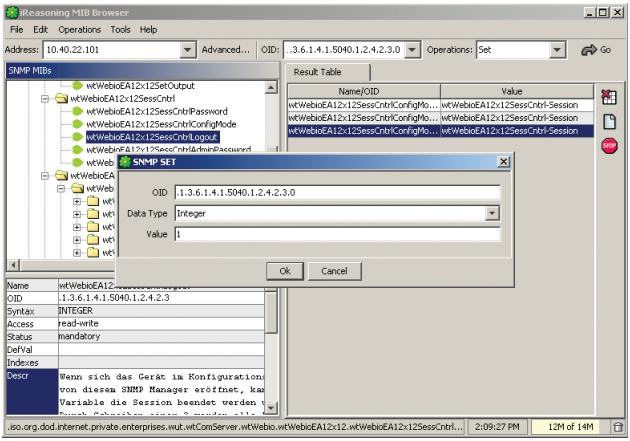
Auch die Grundeinstellungen des Web-IO lassen sich mittels SNMP auslesen und bei gültigem Session-Login auch ändern. Die Private-MIB bildet hierzu alle Parameter ab, die auch über die Browser-Oberfläche konfigurierbar sind. Anders als beim Setzen der Outputs (erfolgt bei gültiger Session sofort) muss Die Session ordnungsgemäß geschlossen werden, damit die Änderungen wirksam werden.

Das geschieht durch Schreiben der Variablen

`wtWebioEA12x12SessCntrlLogout`

Dabei kann über den Wert bestimmt werden wie beendet wird:

1 alle Änderungen werden gespeichert
2 Beenden ohne Speichern




Findet bei geöffneter Session über einen Zeitraum von 5 Minuten keine SNMP-Kommunikation statt, beendet das Web-IO seinerseits die Session und alle Änderungen werden verworfen.

*Das Öffnen einer SNMP-Session hat Vorrang vor einem HTTP-Login. Das bedeutet: Ein User mit Config- oder Administrator-Rechten verliert seinen Browser-Zugriff, sobald eine SNMP-Session geöffnet wird.*

### 5.4.2 OPC - standardisierter Zugriff

OPC (OLE for Process Control) ist eine Softwareschnittstelle zum Zugriff auf Prozessdaten, auf Basis der OLE-Technologie von Microsoft.

Anwendungsprogramme wie z.B. Visualisierungssysteme, die diese Schnittstelle benutzen, heißen OPC-Clients. Auf der anderen Seite der Schnittstelle stehen ihnen OPC-Server gegenüber. Das sind Gerätetreiber, die eine bestimmte Hardware in abstrakter Form als einen Satz von OPC-Variablen darstellen.

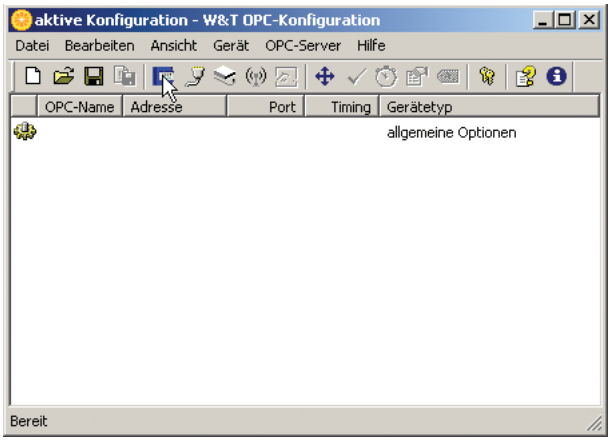
**Die Installation des OPC-Servers**
Den OPC-Server finden Sie auf der dem Web-IO beiliegenden Produkt-CD im Bereich Web-IO Digital.

*Auf unserer Webseite www.WuT.de finden Sie links die„Suche über Artikel-Nr". Geben Sie hier die Artikel-Nr. Ihres Web-IO ein, wählen Sie in dem Feld darunter „Tools" aus und klicken Sie auf „Los". Auf der Seite, die sich dann öffnet, wählen Sie den Link „OPC-Server".*

Zum einfachen Download wurden die benötigten Dateien in einem ZIP-File untergebracht. Speichern Sie die entpackten Dateien in einem beliebigen Verzeichnis Ihrer Festplatte (z.B. C:/Temp) Starten Sie das Setup-Programm in dem Sie z. B. aus dem Startmenü *Ausführen...* wählen und dann eingeben: „C\Temp\Setup". Es installiert und registriert den OPC-Server auf Ihrem Rechner.

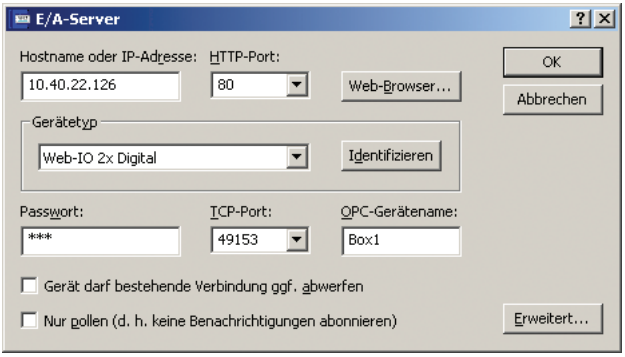
**Deinstallation**
Den OPC-Server können Sie über die Systemsteuerungskomponente „Software" entfernen. Er steht dort in der Liste unter *W&T OPC-Server*

**Konfiguration**
Starten Sie zunächst das Konfigurationsprogramm des OPC-Servers. Bei normaler Installation finden Sie das entsprechend Start Icon auf Ihrer Windows-Oberfläche unter *Start >> Programme >> W&T OPC-Server Version 4 >> OPC-Server konfigurieren*




Klicken Sie auf das Web-IO Icon oder im Menü auf *Gerät >> Neues E/A-Gerät.*

Es öffnet sich folgendes Fenster:




**Hostname oder IP-Adresse**
Tragen Sie hier die IP-Adresse Ihres Web-IO ein. Falls der Adresse über DNS oder einen ähnlichen Namensdienst auch ein Hostname zugeordnet wurde, können Sie auch diesen Namen statt der IP-Adresse verwenden.

**HTTP-Port**
Geben Sie hier den Port ein, den das Web-IO als HTTP-Port verwendet (werksseitig 80).

**TCP-Portnummer**
Muss dem Local Port Eintrag des Web-IO entsprechen (werksseitig 49153)

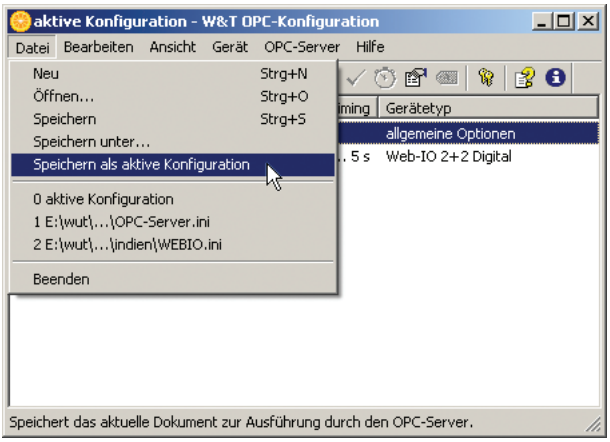
**Passwort**
Geben Sie hier das Operator- bzw. Administrator-Passwort des Web-IO an. Falls Sie sich über das korrekte Passwort unsicher sind, achten Sie bitte nach Beenden des Dialogs auf die Statusmeldung für das Gerät, in der ggf. auf Probleme hingewiesen wird („Passwort erforderlich“ bzw. „falsches Passwort“).

**Gerätetyp**
Wählen Sie an dieser Stelle z.B. Web-IO 2xDigital

**OPC-Gerätename**
Tragen Sie hier den Namen ein, der unter OPC das neu hinzugefügte Gerät im vorderen Teil des OPC-Item Namens wiedergeben soll.

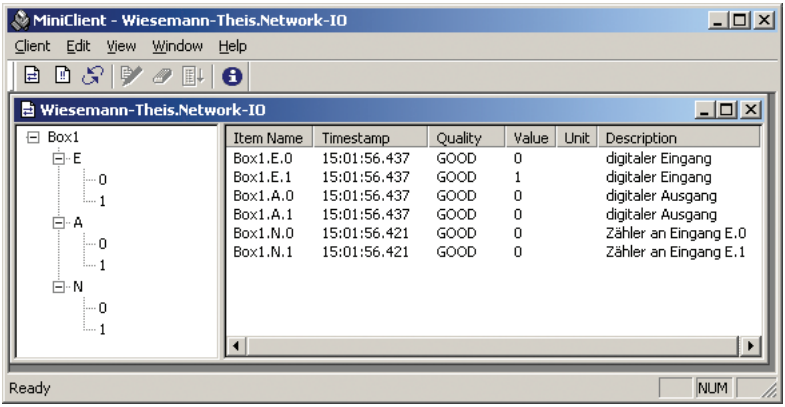
**Nur pollen (keine Benachrichtigungen abonnieren)**
Auf die automatischen Benachrichtigungen zu verzichten, hat den Vorteil, dass der Umfang des Datenverkehrs auf dem Netzwerk nicht von äußeren Ereignissen abhängt.

Abschließend müssen Sie die Konfiguration als aktive Konfiguration speichern. Wählen Sie dazu im Menü *Datei >> Speichern als aktive Konfiguration.*




OPC-Client Applikationen können nun auf das Web-IO zugreifen. Der OPC-Server startet als Dienst und ist unter dem Namen *Wiesemann-Theis.Network-IO* in der OPC-Serverliste zu finden.

Eine Übersicht der in diesem Namensraum verfügbaren Items gibt der OPC-Miniclient der unter *Start >> Programme >> W&T OPC-Server Version 4 >> Minimaler OPC-Client* startbar ist.

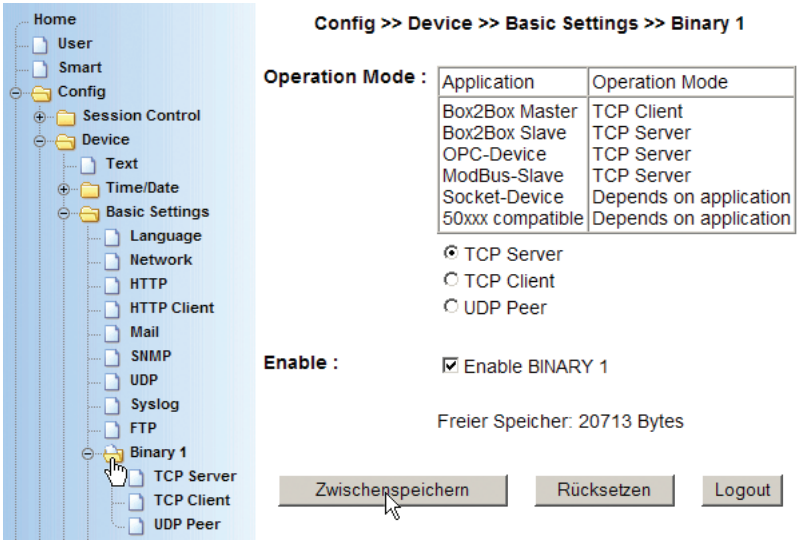



**Web-IO als OPC-Device konfigurieren**

Notwendige Zugriffsrechte: *Administrator*

Wählen Sie im Navigationsbaum des Web-IO *Config >> Device*

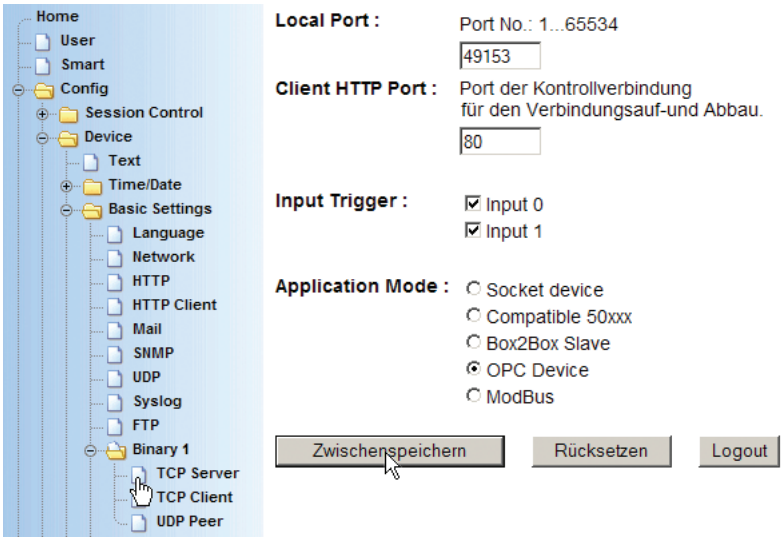
*>> Basic Settings >> Binary 1*




Stellen Sie als *Operation Mode* die Betriebsart *TCP-Server* ein.

Klicken Sie anschließend auf den *Zwischenspeichern*-Button um die Änderungen ans Web-IO zu übertragen.

Wählen Sie nun im Navigationsbaum: *Config >> Device >> Basic Settings >> Binary 1 >> TCP-Server.*




**Local Port**
Wenn Ihnen von Ihrem Netzwerkadministrator keine besonderen Vorgaben gemacht werden, kann der werksseitig eingestellte Port 49153 übernommen werden.

Grund für eine Änderung des werksseitig eingestellten Local Port kann z.B. eine Firewall sein, die nur bestimmte Portzugriffe erlaubt.

*In jedem Fall muss der eingestellte Local Port des Web-IO identisch mit den entsprechenden Einstellungen im OPC-Server sein.*

**Client HTTP Port**
legt den HTTP Port fest, auf den die Steuerverbindung zum OPC-Server aufgebaut werden soll.

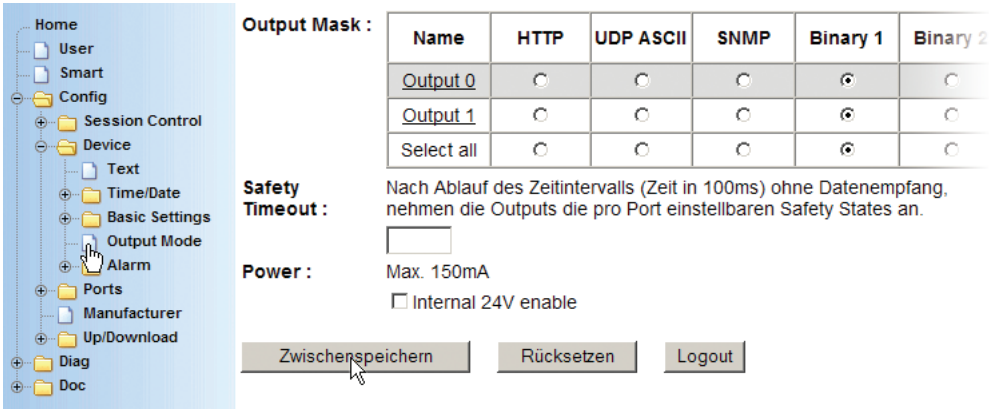
Wenn nicht anders vorgegeben, sollte hier immer Port 80 verwendet werden.

**Input Trigger**
Aktivieren Sie hier die Inputs, die bei Zustandsänderung eine Benachrichtigung an den OPC-Server auslösen sollen.

**Application Mode**
Markieren Sie *OPC Device*

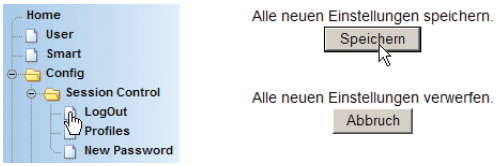
Nachdem alle Parameter eingegeben wurden, bestätigen Sie durch Klick auf den *Zwischenspeichern*-Button.

Wählen Sie nun im Navigationsbaum: *Config >> Device >> Output Mode*




Aktivieren Sie hier die Outputs, die vom OPC-Server gesteuert werden sollen und bestätigen Sie durch Klick auf den *Zwischenspeichern*-Button.

Nun müssen die neuen Einstellungen noch aktiviert werden. Benutzen Sie den *Logout*-Button oder wählen Sie im Menü *Config >> Session Control >> LogOut.*




Nach Klick auf den *Speichern*-Button werden alle Einstellungen im Web-IO aktualisiert und die Startseite im Default User Modus neu aufgebaut. Das Web-IO kann nun vom OPC-Server angesprochen werden.

### 5.4.3 Modbus TCP - standardisierter Zugriff

Modbus TCP ist eine Softwareschnittstelle zum adressbasierten Zugriff auf Prozessdaten. Die W&T Web-IOs arbeiten bei entsprechender Konfiguration als Modbus-Server, die von Modbus-Steuerungen (Clients bzw. Master) kontrolliert werden können.

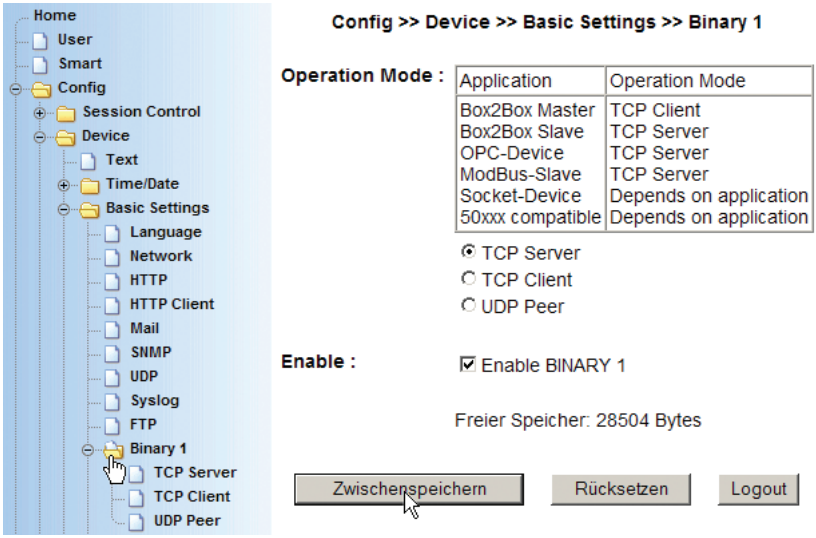
Über den Modbus TCP Zugang können die Zustände von Outputs, Inputs und Countern gelesen werden. Auch Status- und Systemabfragen sind möglich. Darüber hinaus lassen sich die Outputs über Modbus TCP schalten und die Counter löschen.

Als Besonderheit bietet das Web-IO 64 16-Bit-Register, die frei beschreibbar sind und auf die auch vom Browser aus zugegriffen werden kann. Darüber hinaus können über das Beschreiben bestimmter Register Alarme ausgelöst werden, die dann z.B. zum Versand von Emails oder SNMP-Traps führen.

**Konfiguration für den Modbus TCP Zugriff**

Notwendige Zugriffsrechte: *Administrator*

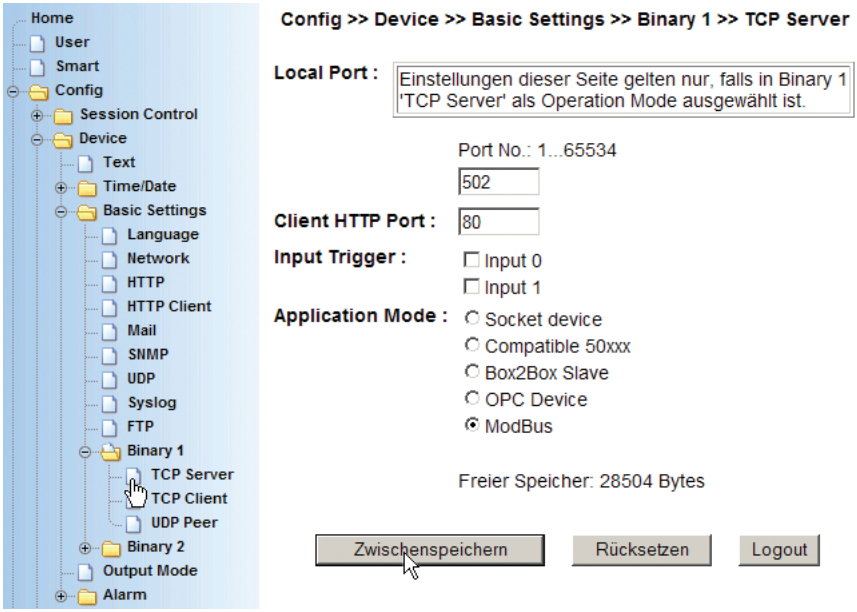
Wählen Sie im Navigationsbaum des Slave Web-IO *Config >> Device >> Basic Settings >> Binary 1*



Stellen Sie als *Operation Mode* die Betriebsart *TCP-Server* ein und aktivieren Sie *Enable Binary*.

Klicken Sie anschließend auf den *Zwischenspeichern*-Button, um die Änderungen ans Web-IO zu übertragen.

Wählen Sie nun im Navigationsbaum: *Config >> Device >> Basic Settings >> Binary 1 >> TCP-Server*.




**Local Port**

Für den Einsatz im normalen Modbus TCP Umfeld muss der lokale Port des Web-IO auf 502 gesetzt werden.

**Client HTTP Port**

Beim Modbus TCP Zugriff wird dieser Port nicht verwendet und kann auf der werksseitigen Vorgabe 80 verbleiben

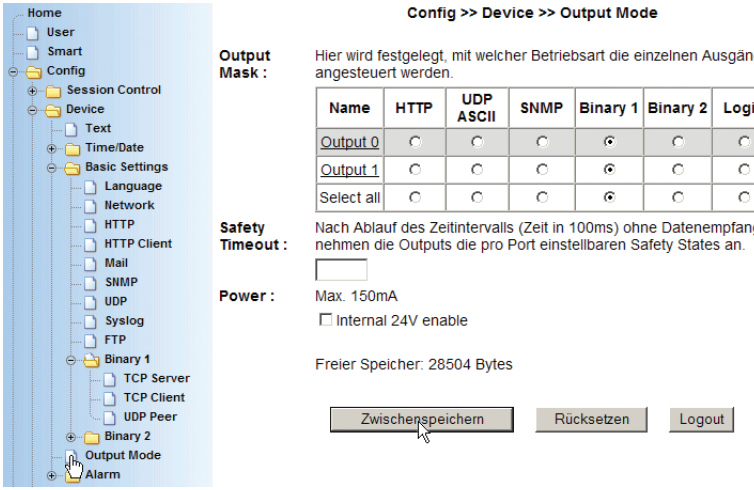
**Input Trigger**

Die *Input Trigger* dürfen für den Modbus TCP Betrieb nicht aktiviert sein.

**Application Mode**
Markieren Sie *Modbus*

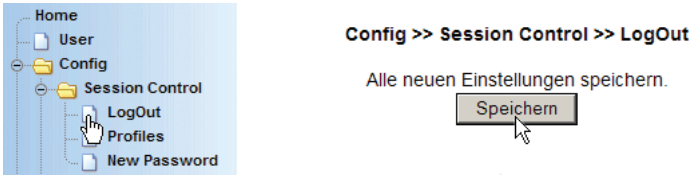
Nachdem alle Parameter eingegeben wurden, bestätigen Sie durch Klick auf den *Zwischenspeichern*-Button.

Wählen Sie nun im Navigationsbaum: *Config >> Device >> Output Mode*




Aktivieren Sie hier die Outputs, die vom Modbus-Client gesteuert werden sollen für Binary 1 und bestätigen Sie mit dem *Zwischenspeichern*-Button.

Nun müssen die neuen Einstellungen noch aktiviert werden. Benutzen Sie den *Logout*-Button oder wählen Sie im Menü *Config >> Session Control >> Logout.*




Nach Klick auf den *Speichern*-Button werden alle Einstellungen im Web-IO aktualisiert und die Startseite im Default User Modus neu aufgebaut. Das Web-IO kann nun vom Modbus

Client angesprochen werden.

**Modbus TCP Kommunikation**
Modbus TCP ist ein Master/Slave Verfahren, bei dem ein Modbus-Master einen Request zum Slave (Web-IO) sendet und der Slave (Web-IO) mit einen Reply antwortet.

Der Datenaustausch zwischen Modbus-Client und Web-IO erfolgt über Modbus TCP Datenpakete.

Das Web-IO verfügt über einen Speicher, aus dem der Modbus-Master lesen bzw. in den er schreiben kann. Dieser Speicher ist in Bereiche aufgeteilt, die bestimmte Eigenschaften des Web-IO repräsentieren.

Es gibt Bereiche für den Status der Inputs, Outputs und Counter. Andere Bereiche geben den Systemstatus wieder oder betreffen die Alarme.

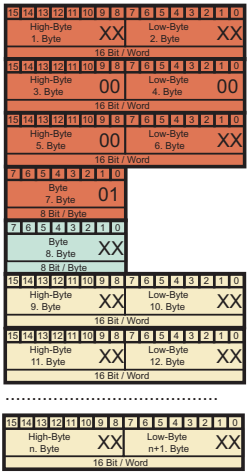
Eine Besonderheit des Web-IO ist ein Speicherbereich, der auf der einen Seite per Modbus-Zugriff frei beschreib- und lesbar ist und auf der anderen Seite über das Web-Interface abgerufen werden kann. Bei entsprechender Webseiten-Programmierung kann ein Modbus-System so Prozessdaten im Browser visualisieren.

Eine weitere Sonderfunktion des Web-IO ist, dass durch Schreiben in einen bestimmten Speicherbereich ein Alarm ausgelöst werden kann.

*Eine Liste mit der genauen Speicheraufteilung folgt im weiteren Verlauf dieses Kapitels.*

Die Modbus Datenpakete bestehen immer aus einem Header, dem Function Code, der Start Address und weiteren Parametern bzw. Registern.

**Aufbau von Modbus TCP Datenpaketen**




| | |
|---|---|
| **Transaction ID** | 16 Bit<br>Wird vom Client je Sendung um 1 hochgezählt |
| **Protocol ID** | 16 Bit<br>Immer = 0x0000 |
| **Length** | 16 Bit<br>Anzahl der Bytes nach dem Header |
| **Unit ID** | 8 Bit<br>Immer = 0x01 |
| **Function Code** | 8 Bit<br>Bestimmt den Zweck der Datensendung |
| **Start Address** | 16 Bit<br>Adresse der zu lesenden/schreibenden Register |
| **Weitere Parameter oder Register** | 8 Bit, 16 Bit oder 32 Bit |

### Transaction Identifier

Dient dazu, die Antwort des Web-IO der Anfrage des Clients zuzuordnen. Der Client zählt dazu normalerweise die ID mit jeder neuen Datensendung um 1 hoch. Das Web-IO gibt immer den empfangenen Wert 1:1 zurück.

### Protocol Identifier

Für die Kommunikation mit dem Web-IO ohne Bedeutung und immer 0x0000

### Length

Anzahl der Bytes, die nach *Length* gesendet werden (gesamte gesendete Bytes - 6).

### Unit identifier

Bei Modbus TCP immer 0x01

### Function Code

Über den *Function Code* wird festgelegt, wie auf den Modbus-Speicher des Web-IO zugegriffen werden soll:

- bitweise
- registerweise (16-bitweise)

und welchen Zweck der Request hat:

- 0x01 Read Coils - lesen einzelner Bits
- 0x02 Read Discret Inputs - lesen einzelner Bits
- 0x03 Read Holding Registers- lesen mehrerer Register
- 0x04 Read Input Register - lesen mehrerer Eingangs Reg.
- 0x05 Write Single Coil - Schreiben eines Bits
- 0x06 Write Single Register - Schreiben nur eines Registers
- 0x07 Read Exception State - Fehlerstatus lesen
- 0x0F Write Multiple Coils - schreiben mehrerer Bits
- 0x10 Write Multiple Register - schreiben mehrerer Register

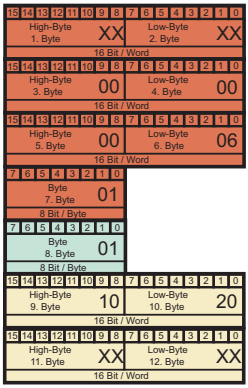
**Start Address**
Über *Start Address* wird festgelegt, auf welchen Bereich des Modbus-Speichers im Web-IO zugegriffen werden soll. Damit wird gleichzeitig festgelegt, auf welche Eigenschaft (Inputs, Outputs, Counter, Alarme .....) zugegriffen wird.

*Eine Liste mit der genauen Speicheraufteilung folgt im weiteren Verlauf dieses Kapitels.*

*Die beiden Parameter* Function Code *und* Start Address *bestimmen, was der gesendete Request beim Web-IO auslösen soll.*

**Weitere Parameter und Register**
Je nach verwendetem *Function Code* können weitere Parameter bzw. Register folgen.

**Function Code 0x01 Read Coils**
Der *Function Code* 0x01 ist zum binären Lesen des Status der Web-IO Outputs vorgesehen.



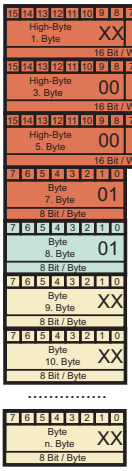

| | |
|---|---|
| **Transaction ID** | 16 Bit<br>Wird vom Client je Sendung um 1 hochgezählt |
| **Protocol ID** | 16 Bit<br>Immer = 0x0000 |
| **Length** | 16 Bit<br>Anzahl der Bytes nach dem Header |
| **Unit ID** | 8 Bit<br>Immer = 0x01 |
| **Function Code** | 8 Bit<br>Bestimmt den Zweck der Datensendung |
| **Start Address** | 16 Bit<br>Adresse der zu lesenden/schreibenden Register |
| **Quantity of Outputs** | 16 Bit<br>Anzahl der Bits die gelesen werden sollen |

**Start Address**
gibt an, ab welcher Speicherstelle Bits (Outputs) gelesen werden sollen.

**Quantity of Outputs**
Hier wird die Anzahl der zu lesenden Bits (Outputs) übergeben.

Über das Senden eines Datenpaketes mit FC 0x01 können ein oder mehrere boolsche Werte (Output-Zustände also 0/1 bzw. ON/OFF) abgefragt werden.

Die Antwort des Web-IO sieht so aus:




| | |
|---|---|
| **Transaction ID** | 16 Bit<br>Wird vom Client je Sendung um 1 hochgezählt |
| **Protocol ID** | 16 Bit<br>Immer = 0x0000 |
| **Length** | 16 Bit<br>Anzahl der Bytes nach dem Header |
| **Unit ID** | 8 Bit<br>Immer = 0x01 |
| **Function Code** | 8 Bit<br>Bestimmt den Zweck der Datensendung |
| **Byte Count** | 8 Bit<br>Anzahl der zurückgeegebenen Bytes<br>(pro angefangene 8 Bit je ein Byte) |
| **Output State** | 8 Bit<br>Die angefragten Bits werden beginnend<br>beim 10.Byte Bit 0 übergeben |

**Bytes Count**
enthält die Anzahl der übertragenen Bytes, welche die angefragten Bits beinhalten. Pro angefangene 8 Bit wird ein Byte gesendet. Das bedeutet: ab 9 Bit werden 2 Bytes gesendet.

**Output State**
Anzahl an Bytes, wie in *Byte Count* übergeben. Das erste State-Byte (10. Byte) enthält beginnend bei Bit 0 die ersten 8 angefragten Bits (Outputs). Wurden weniger als 8 Bits angefragt, werden die ungenutzten Bits mit 0 übergeben. Bei mehr als 8 Bit geht es mit dem 11. Byte weiter.

**Beispiel: Lesen des Status der Outputs 0 und 1**
Output 0 ist OFF, Output 1 ist ON

*Start Address* wird auf 0x1020 gesetzt, *Quantity of Outputs* auf 0x02. (Die Outputs sind ab Adresse 0x1020 abrufbar.)

Das Web-IO antwortet mit *Byte Count* = 1, ein *Output State* = 0x02.

**Function Code 0x02 Read Discrete Inputs**
Der *Function Code* 0x02 ist zum binären Lesen des Status der Web-IO Inputs vorgesehen.

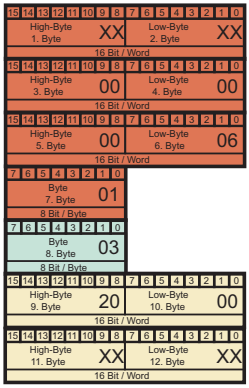
Der Paketaufbau von Anfrage und Antwort mit *Function Code* 0x02 ist identisch mit dem von Function Code 0x01.

**Beispiel: Lesen des Status der Inputs 0 und 1**
Input 0 ist OFF, Input1 ist ON

*Start Address* wird auf 1000 gesetzt,*Quantity of Inputs* auf 2. (Die Inputs sind ab Adresse 0x1000 abrufbar.)

Das Web-IO antwortet mit *Byte Count* = 1, *Input State* = 0x02.

**Function Code 0x03 Read Holding Registers**
Der *Function Code* 0x03 ist zum Lesen von mehreren Registern 16-Bit Werten) vorgesehen. Über FC 0x03 können beim Web-IO je nach verwendeter *Start Address* die Werte von Inputs, Outputs, Counter, usw. abgefragt werden.

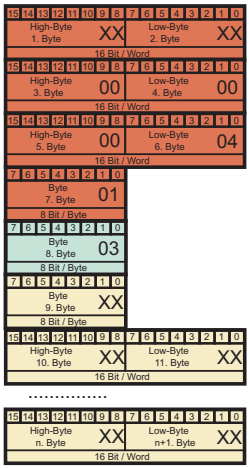



| | |
|---|---|
| **Transaction ID** | 16 Bit<br>Wird vom Client je Sendung um 1 hochgezählt |
| **Protocol ID** | 16 Bit<br>Immer = 0x0000 |
| **Length** | 16 Bit<br>Anzahl der Bytes nach dem Header |
| **Unit ID** | 8 Bit<br>Immer = 0x01 |
| **Function Code** | 8 Bit<br>Bestimmt den Zweck der Datensendung |
| **Start Address** | 16 Bit<br>Adresse der zu lesenden/schreibenden Register |
| **Number of Registers** | 16 Bit<br>Anzahl der Register die gelesen werden sollen |

**Start Address**
gibt an, ab welcher Speicherstelle Register gelesen werden sollen.

**Number of Registers**
Hier wird die Anzahl der zu lesenden Register übergeben.

Das Web-IO antwortet mit folgendem Paket:




| | |
|---|---|
| **Transaction ID** | 16 Bit<br>Wird vom Client je Sendung um 1 hochgezählt |
| **Protocol ID** | 16 Bit<br>Immer = 0x0000 |
| **Length** | 16 Bit<br>Anzahl der Bytes nach dem Header |
| **Unit ID** | 8 Bit<br>Immer = 0x01 |
| **Function Code** | 8 Bit<br>Bestimmt den Zweck der Datensendung |
| **Byte Count** | 8 Bit<br>Anzahl der zurückgeegebenen Register-Bytes |
| **Register Value** | 16 Bit<br>Die angefragten Register werden beginnend beim 10.Byte übergeben |

**Bytes Count**
enthält die Anzahl der als Register übertragenen Bytes (2 Bytes pro 16-Bit Register).

**Register Value**
Ein oder mehrere 16-Bit Registerwerte. Das erste 16-Bit Register beginnt mit dem High-Byte an der Position des 10. Bytes.

Je nach Start Address (ab 0x5000) werden für ein angefragtes 32-Bit Register zwei 16-Bit Werte - also 4 Bytes - übergeben. Auch in diesem Fall beginnt der Wert mit dem höchsten Byte an Position des 10. Byte und das erste Low-Byte liegt auf Position des 13. Byte des Datenpaketes.

**Beispiel:**
**Lesen des Status der Outputs eines Web-IO 2xDigital**
Output 0 ist OFF, Output1 ist ON

*Start Address* wird auf 0x2002 gesetzt,
*Number of Registers* auf 0x01
(Die Outputs sind als Register ab Adresse 0x2002 abrufbar.)

Das Web-IO antwortet mit

*Byte Count* = 0x02,
*Register Value* = 0x0002.

**Function Code 0x04 Read Input Registers**
Der *Function Code* 0x04 ist speziell zum Lesen des Status der Web-IO Inputs als 16-Bit Register vorgesehen.

Der Paketaufbau von Anfrage und Antwort mit Function Code 0x04 ist identisch mit dem von Function Code 0x03.

**Beispiel:**
**Lesen des Status der Inputs eines Web-IO 2xDigital**
Input 0 ist OFF, Input1 ist ON

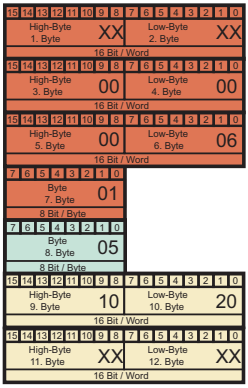
*Start Address* wird auf 0x2000 gesetzt,
*Number of Registers* auf 0x01
(Die Inputs sind als Register ab Adresse 0x2002 abrufbar.)

Das Web-IO antwortet mit

*Byte Count* = 0x02
*Register Value* = 0x0002

### Function Code 0x05 Write Single Coil

Der *Function Code* 0x05 ist zum Setzen eines einzelnen Outputs vorgesehen.




| | |
|---|---|
| **Transaction ID** | 16 Bit<br>Wird vom Client je Sendung um 1 hochgezählt |
| **Protocol ID** | 16 Bit<br>Immer = 0x0000 |
| **Length** | 16 Bit<br>Anzahl der Bytes nach dem Header |
| **Unit ID** | 8 Bit<br>Immer = 0x01 |
| **Function Code** | 8 Bit<br>Bestimmt den Zweck der Datensendung |
| **Output Address** | 16 Bit<br>Adresse des zu schreibenden Registers |
| **Output Value** | 16 Bit<br>Wert der in das Register geschrieben werden soll |

**Output Address**

Über *Output Address* wird indirekt festgelegt, welcher Output geschaltet werden soll. Beginnend bei 0x1020 hat jeder Output eine eigene Adresse.

**Output Value**

Hier wird festgelegt, ob ein Output auf ON oder OFF geschaltet werden soll:

ON = 0xFF00
OFF = 0x0000

Das Web-IO antwortet mit einem exakt gleich aufgebauten Datenpaket.

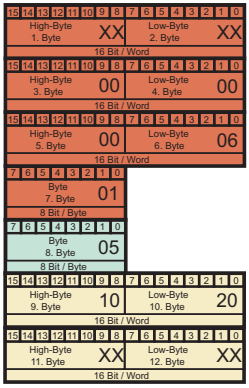
**Beispiel:**
**Output1 eines Web-IO soll auf ON geschaltet werden**
*Output Address* wird auf 0x1021 gesetzt,
*Output Value* auf 0xFF00

Das Web-IO antwortet mit

*Output Address* = 1021
*Output Value* = 0xFF00

### Function Code 0x06 Write Single Register

Der *Function Code* 0x06 arbeitet identisch zum Function Code 0x05 und ist zum Setzen eines beliebigen Registers vorgesehen.




| | |
|---|---|
| **Transaction ID** | 16 Bit<br>Wird vom Client je Sendung um 1 hochgezählt |
| **Protocol ID** | 16 Bit<br>Immer = 0x0000 |
| **Length** | 16 Bit<br>Anzahl der Bytes nach dem Header |
| **Unit ID** | 8 Bit<br>Immer = 0x01 |
| **Function Code** | 8 Bit<br>Bestimmt den Zweck der Datensendung |
| **Register Address** | 16 Bit<br>Adresse des zu schreibenden Registers |
| **Register Value** | 16 Bit<br>Wert der in das Register geschrieben werden soll |

### Register Address

Über *Register Address* wird festgelegt, in welche Speicherstelle geschrieben werden soll.

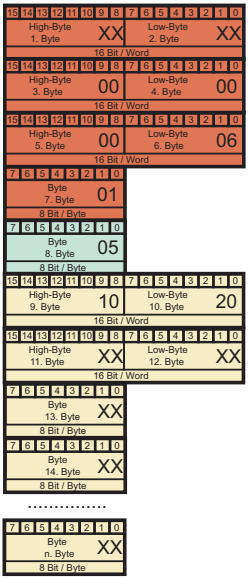
### Register Value

ist ein 16-Bit Registerwert, der in den Modbus-Speicher des Web-IO geschrieben wird.

Das Web-IO antwortet mit einem exakt gleich aufgebauten Datenpaket.

### Function Code 0x0F Write Multiple Coils

Der *Function Code* 0x0F ist für das bitweise Setzen von Outputs vorgesehen.




| | |
|---|---|
| **Transaction ID** | 16 Bit<br>Wird vom Client je Sendung um 1 hochgezählt |
| **Protocol ID** | 16 Bit<br>Immer = 0x0000 |
| **Length** | 16 Bit<br>Anzahl der Bytes nach dem Header |
| **Unit ID** | 8 Bit<br>Immer = 0x01 |
| **Function Code** | 8 Bit<br>Bestimmt den Zweck der Datensendung |
| **Start Address** | 16 Bit<br>Adresse des ersten zu setzenden Outputs |
| **Quantity of Outputs** | 16 Bit<br>Anzahl der zu setzenden Outputs |
| **Byte Count** | 8 Bit<br>Anzahl der Bytes nach ByteCount<br>(pro angefangene 8 Bit je ein Byte) |
| **Output State** | 8 Bit<br>Die Output-Bits werden beginnend<br>beim 14.Byte Bit 0 übergeben |

### Start Address

gibt an, ab welcher Speicherstelle Output-Bits geschrieben werden sollen.

### Quantity of Outputs

Hier wird die Anzahl der zu schreibenden Bits übergeben.

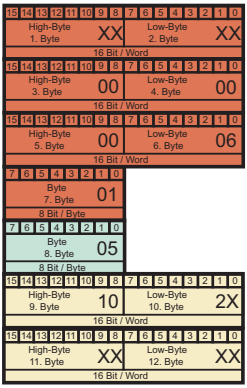
### Bytes Count

enthält die Anzahl der übertragenen Bytes welche die zu setzenden Output-Bits beinhalten. Pro angefangene 8 Bit wird ein Byte gesendet. Das bedeutet ab 9 Bit werden 2 Bytes gesendet.

### Output State

Anzahl an Bytes, wie in *Byte Count* übergeben. Das erste Byte (14. Byte) enthält beginnend bei Bit 0 die ersten 8 zu setzenden Output-Bits (Outputs). Wurden weniger als 8 Bits (Outputs) angegeben, werden die ungenutzten Bits mit 0 übergeben. Bei mehr als 8 Bits geht es mit dem 15. Byte weiter.

Das Web-IO antwortet mit einem Datenpaket mit folgendem Aufbau:




| | |
|---|---|
| **Transaction ID** | 16 Bit<br>Wird vom Client je Sendung um 1 hochgezählt |
| **Protocol ID** | 16 Bit<br>Immer = 0x0000 |
| **Length** | 16 Bit<br>Anzahl der Bytes nach dem Header |
| **Unit ID** | 8 Bit<br>Immer = 0x01 |
| **Function Code** | 8 Bit<br>Bestimmt den Zweck der Datensendung |
| **Start Address** | 16 Bit<br>Adresse des ersten zu setzenden Outputs |
| **Quantity of Outputs** | 16 Bit<br>Anzahl der zu setzenden Outputs |

**Start Address**
gibt an, ab welcher Speicherstelle Output-Bits geschrieben wurden.

**Quantity of Outputs**
Hier wird die Anzahl der geschriebenen Output-Bits übergeben.

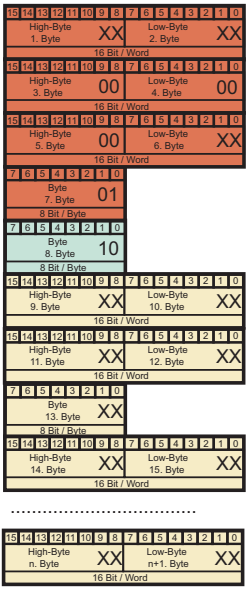
**Beispiel:**
**Output1 eines Web-IO soll auf ON geschaltet werden**
Output 0 soll auf OFF geschaltet werden, Output 1 auf ON

*Output Address* wird auf 1020 gesetzt,
*Quantity of Outputs* auf 0x02
Byte Count auf 0x01
Es wird ein Byte Output Value mit Inhalt 0x02 übergeben.

Das Web-IO antwortet mit

*Output Address* = 1020
*Quantity of Outputs* = 0x0002

**Function Code 0x10 Write Multiple Registers**
Der *Function Code* 0x0F ist für das Schreiben mehrerer 16-Bit Registerwerte vorgesehen.




| | |
|---|---|
| **Transaction ID** | 16 Bit<br>Wird vom Client je Sendung um 1 hochgezählt |
| **Protocol ID** | 16 Bit<br>Immer = 0x0000 |
| **Length** | 16 Bit<br>Anzahl der Bytes nach dem Header |
| **Unit ID** | 8 Bit<br>Immer = 0x01 |
| **Function Code** | 8 Bit<br>Bestimmt den Zweck der Datensendung |
| **Start Address** | 16 Bit<br>Adresse des ersten zu setzenden Outputs |
| **Quantity of Registers** | 16 Bit<br>Anzahl der zu schreibenden Register |
| **Byte Count** | 8 Bit<br>Anzahl der Bytes nach ByteCount |
| **Register Value** | 16 Bit / 32 Bit<br>Die Register werden beginnend beim 14.Byte mit dem ersten Highbyte übergeben |

**Start Address**
gibt an, ab welcher Speicherstelle Register geschrieben werden sollen.

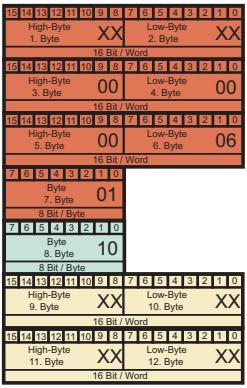
**Quantity of Registers**
Hier wird die Anzahl der zu schreibenden 16-Bit Register übergeben. Beim Schreiben in den 32-Bit Bereich des Web-IO müssen je 32-Bit Wert zwei 16-Bit Register gerechnet werden.

**Bytes Count**
enthält die Anzahl der zu übertragenen Bytes. Für jedes zu sendende 16-Bit Register werden 2 Bytes gerechnet.

**Register Value**
Hier werden die 16-Bit Register übergeben. Das erste High-Byte wird als 14. Byte in das Datenpaket gesetzt. Das erste Low-Byte als 15. Byte usw.

Beim Schreiben in den 32-Bit Bereich des Web-IO (ab Adresse 0x5000) müssen je 32-Bt Wert zwei 16-Bit Register beginnend mit dem höchsten Byte des 32-Bit Wertes geschrieben werden.

Das Web-IO antwortet mit einem Datenpaket mit folgendem Aufbau:




| | |
|---|---|
| **Transaction ID** | 16 Bit<br>Wird vom Client je Sendung um 1 hochgezählt |
| **Protocol ID** | 16 Bit<br>Immer = 0x0000 |
| **Length** | 16 Bit<br>Anzahl der Bytes nach dem Header |
| **Unit ID** | 8 Bit<br>Immer = 0x01 |
| **Function Code** | 8 Bit<br>Bestimmt den Zweck der Datensendung |
| **Start Address** | 16 Bit<br>Adresse des ersten zu setzenden Registers |
| **Quantity of Registers** | 16 Bit<br>Anzahl der geschriebenen Register |

### Start Address

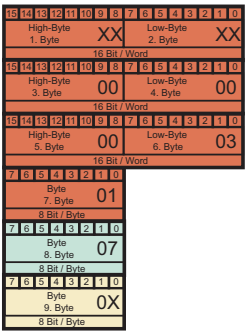
gibt an, ab welcher Speicherstelle die 16-Bit Register geschrieben wurden.

### Quantity of Outputs

Hier wird die Anzahl der geschriebenen 16-Bit Register übergeben.

### Function Code 0x07 Read Exception State

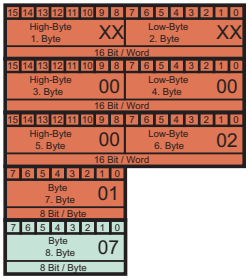
Der *Function Code* 0x07 ist für das Auslesen des Web-IO Fehlerstatus vorgesehen.




| | |
|---|---|
| **Transaction ID** | 16 Bit<br>Wird vom Client je Sendung um 1 hochgezählt |
| **Protocol ID** | 16 Bit<br>Immer = 0x0000 |
| **Length** | 16 Bit<br>Anzahl der Bytes nach dem Header |
| **Unit ID** | 8 Bit<br>Immer = 0x01 |
| **Function Code** | 8 Bit<br>Bestimmt den Zweck der Datensendung |
| **Exception Code** | 8 Bit<br>Web-IO Fehlerstatus |

Vom Master gesendete Pakete mit *Function Code* 0x07 bestehen nur aus Modbus-Header und *Function Code*. Es werden keine weiteren Parameter mitgesendet.

Das Web-IO antwortet mit einem Datenpaket mit folgendem Aufbau:




| | |
|---|---|
| **Transaction ID** | 16 Bit<br>Wird vom Client je Sendung um 1 hochgezählt |
| **Protocol ID** | 16 Bit<br>Immer = 0x0000 |
| **Length** | 16 Bit<br>Anzahl der Bytes nach dem Header |
| **Unit ID** | 8 Bit<br>Immer = 0x01 |
| **Function Code** | 8 Bit<br>Bestimmt den Zweck der Datensendung |

**Exception Code**
beim Web-IO spiegelt der Exception Code einzig den Zustand der LEDs *System Error* und *On Error* wieder.

Bit 0 = 1 entspricht *System Error*
Bit 2 = 1 entspricht *ON Error*

Alle anderen Bits haben beim Web-IO keine Funktion und werden mit 0 übergeben.

**Fehlerhafte Anfragen vom Master**
Sollte der Modbus TCP Master dem Web-IO Anfragen senden, die in Form und Anzahl der Parameter oder hinsichtlich der gewählten Registeradresse für das Web-IO nicht zu verarbeiten sind, sendet das Web-IO ein Modbus Paket zurück, das vom Aufbau her einer Antwort auf *Function Code* 0x07 entspricht.

Das Web-IO addiert in solchen Fällen auf den empfangenen *Function Code* 0x80 auf und setzt diesen Code in das eigene Paket ein.

Der *Exception Code* enthält weitere Details zum Fehler:

0x01 Function Code vom Web-IO nicht unterstützt
0x02 Schreibversuch auf einen Read Only Bereich
0x03 Schreibversuch mit falschen Daten
0x04 Schreiben auf erlaubten Bereich fehlgeschlagen

**Modbus-Adressbereiche für Web-IO Digital**
Alle Adress-Angaben sind hexadezimal zu verstehen.

Beim Web-IO gibt es verschiedene Modbus-Speicherbereiche:

- Bit-Bereich (ab Adresse 0x1000),
- 16Bit-Bereich (ab Adresse 0x2000)
- 32-Bit-Bereich bzw. 2x16-Bit-Bereich (ab Adresse 0x5000)

Die Adressierung erfolgt im Bit-Bereich bitweise, d.h. 1 Bit benötigt eine Adresse. Im 16-Bit- und im 32-Bit-Bereich findet die Adressierung wortweise (2 Byte) statt.

**Die Inputs**
finden sich im Bit-Bereich ab Adresse 0x1000
(Beispiel 57637: 0x1000 und 0x1001),
im 16-Bit-Bereich ab 0x2000 im 16-Bit-Bereich ab 0x2000
(Beispiel 57637: 0x2000),
im 32-Bit-Bereich ab 0x5000.

**Die Outputs**
finden sich im Bit-Bereich ab Adresse 0x1020
(Beispiel 57637: 0x1020 und 0x1021)
im 16-Bit-Bereich ab 0x2002.
im 32-Bit-Bereich ab 0x5002.

**Die Alarme**
finden sich im Bit-Bereich ab Adresse 0x1040
(Beispiel 57637: 0x1040 und 0x1041),
im 16-Bit-Bereich bei 0x2004.
Alarm-Trigger liegen im Bit-Bereich ab Adresse 0x1800.

**Die Counter**
finden sich im 32-Bit-Bereich ab Adresse 0x5004
(Beispiel 57637: 0x5004/0x5005 und 0x5006/0x5007).

Counter-Werte schreiben: Counter können auf beliebige Werte gesetzt werden.

**Der Exception-Status**
findet sich im Bit-Bereich ab Adresse 0x1050,

im 16-Bit-Bereich bei 0x200C (Low Byte).
Alternativ wird der Exception-Status mit dem *Function Code* 0x07 ausgelesen.

**Der Konfigurations-Status**
findet sich im Bit-Bereich bei Adresse 0x1058,
im 16-Bit-Bereich bei 0x200C (High Byte).

**Der Diagnose-Status**
(Anzahl Fehler) liegt im 16Bit-Bereich bei 0x2005,
im 32-Bit-Bereich bei 0x5048.

**Die Diagnose-Status-Bits**
liegen im 16Bit-Bereich ab 0x2006,
im 32-Bit-Bereich ab 0x5048.
Beim Web-IO Digital sind 84 Fehlermeldungen möglich.

*Eine Liste der möglichen Fehlermeldungen finden Sie im Anhang.*

**Identifizierung des Gerätes**
erfolgt über Seriennummer (ab 0x6000) und Mac-Adresse (ab 0x6004).

**Virtuelle Register für Browser-Interaktion**
Der Speicherbereich, den das Gerät für die Übergabe an Web-Anwendungen zur Verfügung stellt, liegt ab Adresse 0x7000.

**Lesen von nicht unterstützen Registern**
Beim Lesen von Daten (Speicherbereichen), die nicht für das Gerät definiert wurden, gibt das Gerät „0“ zurück.

**Modbus - virtuelle Register**
Das Web-IO stellt 64 virtuelle 16-Bit Register zur Verfügung, in die vom Modbus Master beliebige Werte geschrieben werden können (High Byte first). Das Schreiben in diese Register löst im Web-IO keine speziellen Aktionen aus.

Der virtuelle Speicher dient vielmehr zur Übergabe von Modbus- Prozessdaten an Web-Anwendungen.

Über einen HTTP Request

```
modbusreg?PW=<password>&
```

können die 64 Register (128Byte) von Web-Anwendungen abgerufen werden.

Das Web-IO antwortet mit

```
modbus;<High Byte1>;<Low Byte1>;<High Byte2>;<Low Byte2>;<High Byte3>;.....
```

Das heißt, alle 64 Register (128Byte) werden Byte für Byte mit Semikolon getrennt hinter dem Wort „modbus“ ausgegeben.

Über JavaScript und Programmiertechniken wie AJAX kann so eine Prozessvisualisierung im Browser realisiert werden.

Im einfachsten Fall können die virtuellen Register auf der werksseitig vorhandenen User-Seite des Web-IO tabellarisch angezeigt werden (nur wenn die Betriebsart Modbus im Web-IO aktiviert wurde).

**Web-IO User Site**
**WEBIO-05372A**
Von der Klemme direkt aufs Netzwerk

Logout

| Inputs | | | | Outputs | | |
|---|---|---|---|---|---|---|
| Name | State | Counter | Clear | Name | State | Switch |
| Input 0 | ON | 0 | Clear | Output 0 | OFF | ON OFF |
| Input 1 | ON | 0 | Clear | Output 1 | OFF | ON OFF |

**Free available Modbus Memory**

| Addr. | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7000 | 26 38 | FE 12 | AF 45 | 20 16 | 39 45 | FC D3 | 09 CD | 23 FA | 16 29 | F3 D7 | FD 2A | 45 9F | 35 2F | DA 29 | 58 39 | F2 D1 |
| 7010 | 16 29 | F3 D7 | FD 2A | 45 9F | 35 2F | DA 29 | 58 39 | F2 D1 | 23 4E | FC 56 | 01 29 | 56 39 | 23 E4 | F1 AC | CD CD | 45 91 |
| 7020 | 23 4E | FC 56 | 01 29 | 56 39 | 23 E4 | F1 AC | CD CD | 45 91 | 12 AC | 56 FF | 92 73 | 5F 2A | FA DE | 34 1D | 00 02 | 1F A1 |
| 7030 | 12 AC | 56 FF | 92 73 | 5F 2A | FA DE | 34 1D | 00 02 | 1F A1 | 00 00 | 00 00 | 00 00 | 00 00 | 00 00 | 00 00 | 00 00 | 00 00 |

Show as: ○ 8Bit ◉ 16Bit ○ 32Bit

### Modbus-Alarm-Auslösung

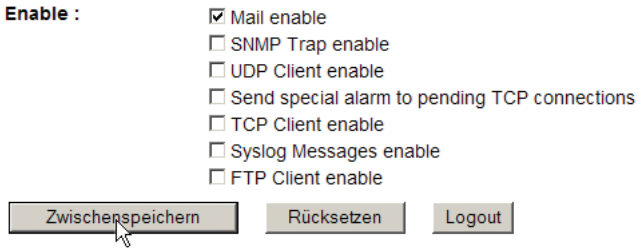
Über ein Alarm-Trigger-Bit im Bit-Speicherbereich können einzelne Alarme ausgelöst werden. Die Bits liegen ab der Adresse 0x1800.

Bei den Modellen 57630, 57631 und 57634 stehen 12

Alarme zur Verfügung. Bei Modell 57637 sind es 4 Alarme.

In der Konfigurationsoberfläche im Browser wir z.B. unter *Config >> Device >> Alarm >> Alarm1* gewählt, auf welchem Weg der Alarm versendet werden soll. Hier z.B. per Email.

Notwendige Zugriffsrechte: *Administrator*

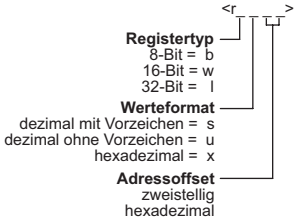



Bei einem Alarm, der über Modbus ausgelöst wird, sollten keine anderen Trigger (*Input Trigger, Output Trigger, System Trigger*) aktiviert sein.

*Die restliche Konfiguration der Alarme erfolgt ganz normal, wie im nachfolgenden Kapitel „Email und mehr - Alarm bei Zustandsänderung“ beschrieben.*

Mit speziellen Tags kann beim Alarmieren (Mail, FTP, SNMP) auch ein Abbild des virtuellen Speichers in die Alarmmeldung mit eingebettet werden. Das Tag steht sozusagen als Platzhalter für den Inhalt eines ausgewählten Registers

Die Tags erlauben die Darstellung in 8-, 16-, oder 32-bitweis. Dabei kann vorgegeben werden, ob der adressierte Wert dezimal oder hexadezimal ausgegeben wird. Darüber hinaus wird der Speicheroffset ab Adresse 0x7000 angegeben.



Beispiel: An Speicherstelle 7004 wurde per Modbus der 16-Bit Wert 0x4711 geschrieben.

Der Alarmtext wurde im Web-IO folgendermaßen konfiguriert:

„<t> - Register 4 hat folgenden Wert: <rwx04>“

Die empfangene Alarmmeldung würde so aussehen:

„2011-02-22 10:34:21 - Register 4 hat folgenden Wert: 4711“

*Alle Details und Möglichkeiten zu den Web-IO Alarmmeldungen lesen Sie bitte im nachfolgenden Kapitel „Email und mehr - Alarm bei Zustandsänderung“.*

## Modbus Speicheraufteilung

### Bit-Bereich:

| Adresse (hexadez.) | Beschreibung | Speicher Modell | Länge (Byte) | Bits lesen mit FC | Reg. lesen mit FC | Bits schreiben mit FC | Reg. schreiben mit FC |
|---|---|---|---|---|---|---|---|
| 1000 | Input 0 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1001 | Input 1 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1002 | Input 2 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1003 | Input 3 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1004 | Input 4 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1005 | Input 5 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1006 | Input 6 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1007 | Input 7 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1008 | Input 8 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1009 | Input 9 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 100A | Input1 0 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 100B | Input 11 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1020 | Output 0 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1021 | Output 1 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1022 | Output 2 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1023 | Output 3 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1024 | Output 4 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1025 | Output 5 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1026 | Output 6 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1027 | Output 7 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1028 | Output 8 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1029 | Output 9 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 102A | Output 10 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 102B | Output 11 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1040 | Alarmstatus 1 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1041 | Alarmstatus 2 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1042 | Alarmstatus 3 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1043 | Alarmstatus 4 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1044 | Alarmstatus 5 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1045 | Alarmstatus 6 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1046 | Alarmstatus 7 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1047 | Alarmstatus 8 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1048 | Alarmstatus 9 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1049 | Alarmstatus 10 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 104A | Alarmstatus 11 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 104B | Alarmstatus 12 | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1060 | Exception-Status | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1068 | Konfig.-Status | Bit | 1 | 0x01, 0x02 | - | - | - |
| 1800 | Alarm-Trigger 1 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1801 | Alarm-Trigger 2 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1802 | Alarm-Trigger 3 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1803 | Alarm-Trigger 4 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1804 | Alarm-Trigger 5 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1805 | Alarm-Trigger 6 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1806 | Alarm-Trigger 7 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1807 | Alarm-Trigger 8 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1808 | Alarm-Trigger 9 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 1809 | Alarm-Trigger 10 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 180A | Alarm-Trigger 11 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |
| 180B | Alarm-Trigger 12 | Bit | 1 | 0x01, 0x02 | - | 0x05 | 0x0F |

*Bitte beachten Sie, dass je nach verwendetem Web-IO nicht die gesamte Breite der Inputs, Outputs, Counter oder Alarme zur Verfügung steht.*

## 16- und 32-Bit-Bereich:

| Adresse (hexadez.) | Beschreibung | Speicher Modell | Länge (Byte) | Bits lesen mit FC | Reg. lesen mit FC | Bits schreiben mit FC | Reg. schreiben mit FC |
|---|---|---|---|---|---|---|---|
| 2000 | Inputs 0 - 11 | 16-Bit | 2 | - | 0x03, 0x04 | - | - |
| 2002 | Outputs 0 - 11 | 16-Bit | 2 | - | 0x03, 0x04 | - | - |
| 2004 | Alarmstatus 1 - 12 | 16-Bit | 2 | - | 0x03, 0x04 | - | - |
| 2006 | Diagnose Fehleranzahl | 16-Bit | 2 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 2007 | Diagnosestatus 0 - 15 | 16-Bit | 2 | - | 0x03, 0x04 | - | - |
| 2008 | Diagnosestatus 16 - 31 | 16-Bit | 2 | - | 0x03, 0x04 | - | - |
| 2009 | Diagnosestatus 32 - 47 | 16-Bit | 2 | - | 0x03, 0x04 | - | - |
| 200A | Diagnosestatus 48 - 63 | 16-Bit | 2 | - | 0x03, 0x04 | - | - |
| 200B | Diagnosestatus 64 - 79 | 16-Bit | 2 | - | 0x03, 0x04 | - | - |
| 200C | Diagnosestatus 80 - 95 | 16-Bit | 2 | - | 0x03, 0x04 | - | - |
| 200D | Exception/Konf.-Status | 16-Bit | 2 | - | 0x03, 0x04 | - | - |
| 5000 | Inputs 0 - 11 | 32-Bit | 4 | - | 0x03, 0x04 | - | - |
| 5002 | Outputs 0 - 11 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 5004 | Alarmstatus 1 - 12 | 32-Bit | 4 | - | 0x03, 0x04 | - | |
| 5006 | Counter 0 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 5008 | Counter 1 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 500A | Counter 2 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 500C | Counter 3 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 500E | Counter 4 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 5010 | Counter 5 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 5012 | Counter 6 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 5014 | Counter 7 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 5016 | Counter 8 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 5018 | Counter 9 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 501A | Counter 10 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 501C | Counter 11 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 504A | Diagnose Fehleranzahl | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 504C | Diagnosestatus 0 - 31 | 32-Bit | 4 | - | 0x03, 0x04 | - | - |
| 504E | Diagnosestatus 32 - 63 | 32-Bit | 4 | - | 0x03, 0x04 | - | - |
| 5050 | Diagnosestatus 64 - 95 | 32-Bit | 4 | - | 0x03, 0x04 | - | - |
| 7000 | virtuelles Register 0 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 7002 | virtuelles Register 1 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 7004 | virtuelles Register 2 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 7006 | virtuelles Register 3 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 7008 | virtuelles Register 4 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 700A | virtuelles Register 5 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 700C | virtuelles Register 6 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 700E | virtuelles Register 7 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 7010 | virtuelles Register 8 | 32-Bit | 4 | - | 0x03, 0x04 | | 0x06, 0x10 |
| ..... | virtuelles Register 9 - 23 | 32-Bit | 4 | | | | |
| 702E | virtuelles Register 23 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 7030 | virtuelles Register 24 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 7032 | virtuelles Register 25 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 7034 | virtuelles Register 26 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 7036 | virtuelles Register 27 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 7038 | virtuelles Register 28 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 703A | virtuelles Register 29 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 703C | virtuelles Register 30 | 32-Bit | 4 | - | 0x03, 0x04 | - | 0x06, 0x10 |
| 703E | virtuelles Register 31 | 32-Bit | 4 | - | 0x03, 0x04 | | 0x06, 0x10 |

*Bitte beachten Sie, dass je nach verwendetem Web-IO nicht die gesamte Breite der Inputs, Outputs, Counter oder Alarme zur Verfügung steht.*

## 5.5 Email und mehr - Alarm bei Zustandsänderung

Im Web-IO können je nach Modell bis zu 12 verschiedene Alarme festgelegt werden. Abhängig von konfigurierbaren Alarmbedingungen können Alarmmeldungen ausgegeben werden. Dazu stehen verschiedene Netzwerkprotokolle zur Verfügung:

- Mail (SMTP)
- SNMP
- Syslog
- UDP
- TCP (über bestehende Verbindung)
- TCP Client
- FTP Client

Neben den Alarmmeldungen versendet das Web-IO Alarmclear-Meldungen, wenn die Alarmbedingungen nicht mehr erfüllt sind, also der Normalzustand wieder vorliegt.

### 5.5.1 Festlegen der Alarmbedingungen

Notwendige Zugriffsrechte: *Administrator*

Die Alarmbedingungen können in verschiedenen Trigger-Gruppen festgelegt werden.

- Input Trigger
- Output Trigger
- System Trigger

Innerhalb eines Alarmes sollten nicht die Bedingungen aus unterschiedlichen Trigger Gruppen gemischt werden.

*Machen Sie das Auslösen eines Alarms entweder vom Status der Inputs oder vom Status der Outputs oder von einem der System Trigger abhängig!*

Es müssen zwei grundsätzliche Arten von Alarm-Triggern unterschieden werden:

- Trigger mit Alarmzustand und Normalzustand
- Trigger als Einzelereignis

**Trigger mit Alarmzustand und Normalzustand**
Triggerbedingung können die Zustände ON oder OFF der In- bzw. Outputs, aber auch der System Trigger Load Control sein.

Wird z.B. für einen Input ON als Trigger gewählt, wird die Alarmmeldung gesendet, wenn dieser Input auf ON geht. Fällt dieser Input zurück auf OFF, sendet das Web-IO die Alarmclear-Meldung.

*Fällt eine Alarmbedingung zurück in den Normalzustand, bevor das Web-IO die Alarmmeldung senden konnte, wird nur die Alarmclear-Meldung gesendet*

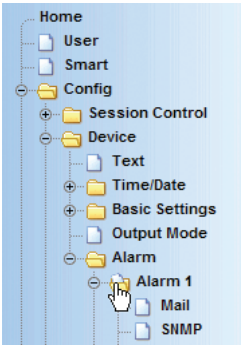
Man kann also sagen, das diese Alarme immer mit einer Alarmclear Meldung abgeschlossen werden.

**Trigger als Einzelereignis**
Hierzu zählen System Trigger wie Kalt- und Warmstart, Timerinterval und die Einstellung ANY bei den Inputs und Outputs.

Zu diesen einmaligen Ereignissen gibt es keinen Normalzustand im eigentlichen Sinne. Da Alarme immer mit einer Alarmclear-Meldung abgeschlossen werden sollten, löst diese Art von Trigger immer nur eine Alarmclear-Meldung aus.

*Es gibt keine Alarmmeldung sondern nur Alarmclear-Benachrichtigungen.*

**Alarm abhängig vom Status der Inputs und Outputs**
Wählen Sie im Navigationsbaum *Config >> Device >> Alarm >> Alarm1* wenn Sie z.B. die Alarmbedingungen für Alarm 1 bearbeiten möchten.




Unter *Input Trigger* und *Output Trigger* kann jeweils festgelegt werden:

1. welche Inputs oder Outputs für diesen Alarm überhaupt auf ihren Status überprüft werden sollen
2. bei welchem Zustand - ON, OFF oder ANY (Flankenwechsel) - die Bedingung erfüllt sein soll

Die Triggerbedingungen der Blöcke Input Trigger, Output Trigger, Time Trigger, Load Control, Cold Start und Warm Start sollten innerhalb eines Alarms nicht gemischt genutzt werden.

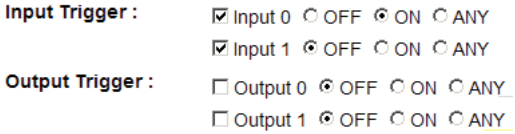
Alle Inputs/Outputs, die markiert sind, müssen die eingestellte Bedingung erfüllen, damit der Alarm ausgelöst wird (UND-Verknüpfung).

*Sind mehrere Inputs/Outputs mit der Bedingung ANY konfiguriert, müssen diese zeitgleich schalten, damit es zu einem Alarm kommt.*

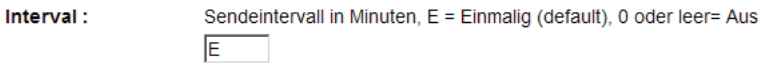
Alarme, die durch ANY getriggert werden, lösen nur eine Alarmclear-Meldung aus!

**Ein Beispiel:**
Ein Alarm soll ausgelöst werden, wenn Input 0 den Status ON hat, und an Input 1 kein Signal also OFF anliegt.

Die Konfiguration müsste in diesem Fall so aussehen:



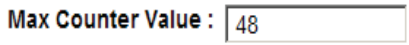

Zusätzlich muss festgelegt werden, ob ein Alarm nur einmalig bei Eintreten der Alarmbedingung ausgelöst wird. Alternativ kann ein Intervall angegeben werden, in dem der Alarm wiederholt wird, solange die Alarmbedingung ansteht.




Im Feld *Interval* kann entweder *E* für einmalig oder ein Intervall in Minuten angegeben werden. Es sind theoretisch Werte zwischen 1 und 9999 Minuten möglich. Wird das Feld nicht oder mit „0“ ausgefüllt, findet keine Alarmierung statt.

**Alarm bei erreichtem Counter-Wert**

Alternativ zum Zustand der Inputs und Outputs kann auf das Erreichen eines Zählerstandes hin ein Alarm ausgelöst werden. Dazu muss im Feld *Max Counter Value* ein Wert zwischen 1- 2,1 Milliarden eingegeben werden. Erreicht einer der aktivierten Inputs diesen Counter-Wert wird Alarm gegeben.



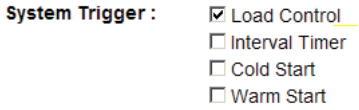

*Counter und Inputmuster schließen sich gegenseitig als Alarmbedingung aus. Sobald unter* Max Counter Value *ein Wert eingetragen ist, wird der Zustand der Inputs nicht mehr als Alarmbedingung gewertet, sondern nur noch das Erreichen des Zählerstandes überwacht.*

**Alarm bei Überlast (nur 57630/57631)**

Die Outputs des Web-IO Digital haben eine thermische Überlastkontrolle. Wird eine Überlast an einem Output erkannt, wird dieser umgehend abgeschaltet. Das gleiche gilt bei komplett fehlender Last.

*Ach der Über- bzw. Unterlastfall kann als Alarmbedingung gewählt werden. Aktivieren Sie hierzu unter* Config >> Device

>> Alarm >> Alarm1, System Trigger *den Punkt* Load Control.

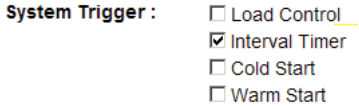



*Im Falle einer Überlast kann der entsprechende Output nach Beseitigung der Ursache durch Aufruf von* Diag >> Report *im Navigationsbaum und Klick auf den* Report Löschen- *Button wieder freigeschaltet werden.*

Beim Freischalten des betroffenen Output wird eine Alarmclear-Meldung gesendet.

**Zeitgesteuerter Alarm**

Durch Aktivierung von *Intervall Timer* kann das Web-IO dazu veranlasst werden - unabhängig von Input- und Output-Status - zu festgelegten Zeiten Alarm zu geben.




Auf diese Weise kann z.B. eine Keep-Alive-Überwachung realisiert werden.

Zu welchen Zeiten Alarm ausgelöst wird, lässt sich im Bereich *Time Trigger* festlegen.

**Time Trigger :** Uhrzeitgesteuerte Ausgabe des Alarms

| Feld | Eingabe [ Zahl * , -] | möglicher Zahlenbereich |
|---|---|---|
| **Minute** | 0-59/15 | 0-59 |
| **Stunde** | 8-17 | 0-23 (0 ist Mitternacht) |
| **Monatstag** | 1-31 | 1-31 |
| **Monat** | 1-12 | 1-12 |
| **Wochentag** | 1-5 | 0-6 (0 ist Sonntag) |

Die Syntax der Zeiteinstellung ist an das in Linux/Unix verwendete Cron-Format angelehnt. In ein Feld können mehrere Angaben mit Komma getrennt eingegeben werden.

Das „*“ steht für jeden zulässigen ganzzahligen Wert. Beispiel: * im Feld Minute gibt minütlich einen Alarm, wenn auch die Bedingungen in den anderen Feldern erfüllt sind.

Wird dem eigentlichen Wert oder Wertebereich ein „/“ nachgestellt kann dahinter eine Zykluswert angegeben werden. Beispiel: 0-59/15 im Feld Minute ergibt alle fünfzehn Minuten einen Alarm, wenn auch die Bedingungen in den anderen Feldern erfüllt sind.

*Wenn mit zyklischen Alarmen gearbeitet werden soll, darf vor dem „/“ nicht mit Wildcards „*“ gearbeitet werden.*

Mit „-“ können Zeitspannen als Trigger bestimmt werden. Beispiel: 8-17 im Feld Stunde bedeutet stündlich zwischen 8 Uhr und 17 Uhr.

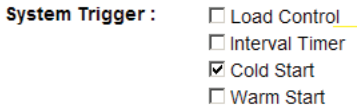
Die ausgefüllten Felder bedeuten also jeden Monat, an jedem Montag bis Freitag, zwischen 8 Uhr und 17 Uhr alle vollen fünfzehn Minuten ein Alarm.

*Es macht keinen Sinn, im Eingabefeld* Interval *etwas anderes als „E“ einzugeben, wenn mit* Time Trigger *gearbeitet wird!*

*Bei Timer-gesteuerten Alarmen erfolgt die Alarmierung durch die Alarmclear-Meldung*

**Alarm bei Kaltstart**

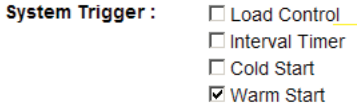
Durch Aktivierung von Cold Start wird das Web-IO veranlasst, nach einem Hardware- bzw. Power-ON- Reset Alarm zu geben.




*Bei Kaltstart gesteuerten Alarmen erfolgt die Alarmierung durch die Alarmclear-Meldung*

**Alarm bei Warmstart**

Durch Aktivierung von *Warm Start* wird das Web-IO veranlasst, nach einem Software-Reset, z.B. durch Connect auf TCP-Port 8888, Alarm zu geben.



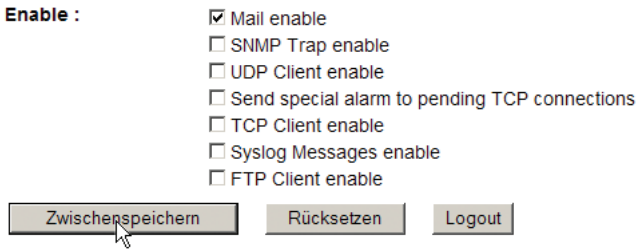

Nachdem die Alarmbedingungen festgelegt wurden, kann ausgewählt werden, auf welchem Weg ein Alarm ausgegeben werden soll.

*Bei Warmstart gesteuerten Alarmen erfolgt die Alarmierung durch die Alarmclear-Meldung*

### 5.5.2 E-Mail-Alarm - SMTP

Für die Alarmierung über E-Mail muss unter *Config >> Device >> Alarm >> Alarm1* zusätzlich zu den Alarmbedingungen *Mail Enable* aktiviert werden.

Notwendige Zugriffsrechte: *Administrator*




Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

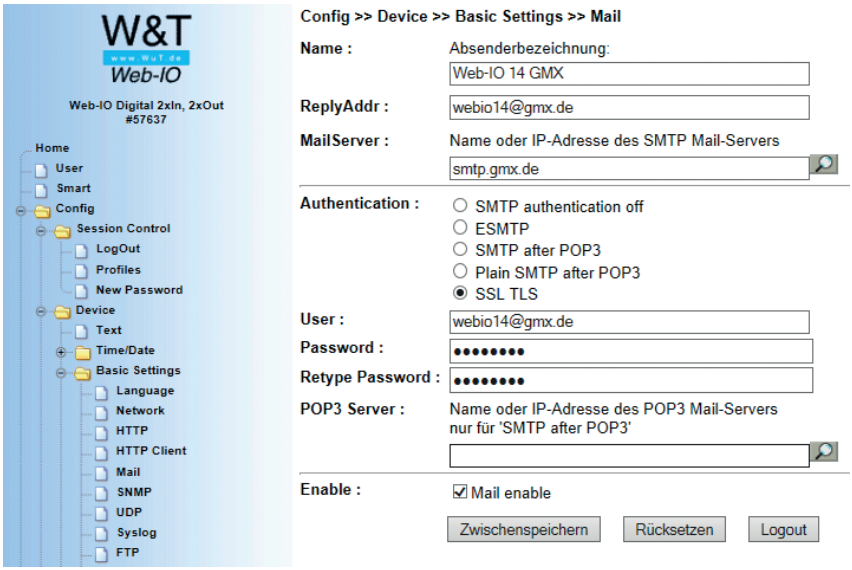
**Grundsätzliche Einstellungen für den E-Mail Versand**

Notwendige Zugriffsrechte: *Administrator*

Damit das Web-IO überhaupt E-Mail versenden kann, müssen

einige Grundeinstellungen vorgenommen werden.

Wählen Sie hierzu im Navigationsbaum *Config >> Device >> Basic Settings >> Mail:*




Füllen Sie die Felder entsprechend Ihrer Applikation aus:

**Name**
Name, der bei versendeten E-Mails als Absender eingetragen wird.

**Reply Address**
E-Mail Adresse die - wenn der Empfänger einer Alarm-Mail im Mail-Client *Antworten* auswählt - als Ziel-Adresse in die abgehende E-Mail eingetragen wird. Da das Web-IO selber nur E-Mail versenden, aber nicht empfangen kann, sollte hier z.B. die E-Mail Adresse des Mitarbeiters eingetragen werden, der das Web-IO technisch betreut. Einige Mail Server versenden Emails aus Sicherheitsgründen nur dann, wenn die Reply Adresse zu einem auf dem Mail Server existierenden Postfach gehört.

**Mail Server**
Hier muss der Host-Name oder die IP-Adresse des SMTP-Servers eingetragen werden, über den das Web-IO seine Alarmmails verschicken soll.

**Authentification**
An dieser Stelle kann ausgewählt werden, ob mit einer E-Mail Authentifizierung gearbeitet werden soll. Je nach Provider-Vorgabe kann für das Web-IO ein Login mit ESMTP, SMTP after POP3 oder wie heute meist üblich, mit SSL-Verschlüsselung konfiguriert werden.

**User**
Tragen Sie hier den Usernamen für das benutzte E-Mail-Konto ein

**Password / Retype Password**
Hier muss das für das E-Mail-Konto vergebene Passwort eingetragen werden.

**POP3 Server**
Wenn mit SMTP after POP3 gearbeitet wird, muss hier der Name oder die IP-Adresse des POP3-Servers eingetragen werden.

**Enable**
Setzen Sie *Mail enable* um den E-Mailversand zu aktivieren.

*Wird der Name des Mail-Servers bzw. des POP3-Servers angegeben, muss in den Netzwerkeinstellungen unbedingt ein DNS-Server angegeben sein.*

Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

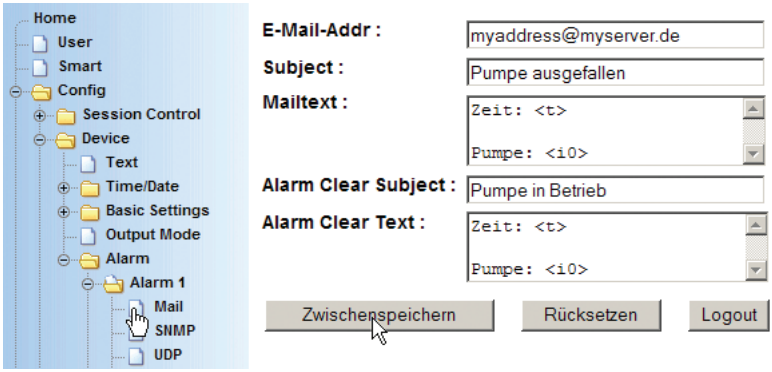
**Festlegen von E-Mail Empfänger, Betreff und Text**

Notwendige Zugriffsrechte: *Administrator*

Unter *Config >> Device >> Alarm>> Alarm1 >> Mail* kann nun festgelegt werden, an wen im Falle eines Alarmes die E-Mail

*Je nach dem, welcher Provider und welche Authentification gewählt wird, nutzen die Mail Server ggf. vom Standard abweichende Portnr. für den TCP-Zugang. Die Portnr. kann in diesem Fall mit einem Doppelpunkt getrennt hinter der Server-Adresse angegeben werden*
*z.B. smtp.gmx.de:587*

gesendet werden soll, was in der Betreffzeile erscheint und welchen Text die E-Mail beinhaltet.




**E-Mail Addr.**
geben Sie in diesem Feld die E-Mail Adresse ein, an die eine E-Mail gesendet werden soll, wenn die Alarmbedingungen erfüllt sind. Wenn mehrere Empfänger die E-Mail erhalten sollen, müssen die E-Mail Adressen mit Semikolon getrennt werden. z.B. user1@domain.de; user2@domain.de; ...

**Subject**
Hier wird die Betreffzeile des Alarm-Mails eingegeben, die bei Auftreten der konfigurierten Alarmbedingung gesendet wird.

**Mailtext**
kann einen frei formulierten Text beinhalten, der bei Auftreten der konfigurierten Alarmbedingung gesendet wird.

**Alarm Clear Subject**
Sobald die konfigurierte Alarmbedingung nicht mehr vorliegt, wird eine Alarmclear-Meldung mit der hier eingetragenen Betreffzeile verschickt.

**Alarm Clear Text**
**Wie beim Mailtext kann hier ein frei formulierter Text für die Alarmclear-Meldung eingetragen werden.**

Zusätzlich können in der Betreffzeile und im Mailtext Alarm-Variable eingesetzt werden. Das sind Platzhalter für Zustand und Namen der einzelnen Inputs und Outputs,

sowie für die Zählerstände der Counter. Darüber hinaus können Platzhalter für Datum und Device Name eingesetzt werden.

Die möglichen Variablen können der folgenden Tabelle entnommen werden:

| Alarm Variable | Beschreibung |
|---|---|
| <dn> | Device Name (siehe: Config>>Device>>Text) |
| <i> | Zustand der Inputs als Bitmuster in hexadezimaler Schreibweise |
| <i**x**> | Zustand des Inputs Nr. **x** (ON/OFF), |
| <in**x**> | Name des Inputs Nr. **x** |
| <o> | Zustand der Outputs als Bitmuster in hexadezimaler Schreibweise |
| <o**x**> | Zustand des Outputs Nr. **x** (ON/OFF), |
| <on**x**> | Name des Outputs Nr.**x** |
| <c**x**> | Zählerstand des Counters Nr. **x** in dezimaler Schreibweise |
| <t> | Zeitstempel mit Datum und Uhrzeit im Format: TT.MMM.JJJJ hh:mm:ss |
| <$y> | Jahr im Format "JJJJ" |
| <$m> | Monat im Format "MM" |
| <$d> | Tag im Format "TT" |
| <$h> | Stunde im Format "hh" |
| <$i> | Minuten im Format "mm" |
| <$s> | Sekunden im Format "ss" |
| | **x** kann je nach Modell zwischen 0 und 23 liegen |

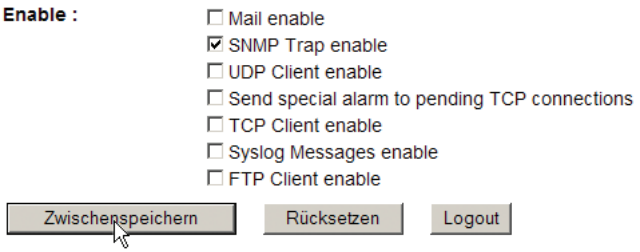
Die Mail-Tags wurden durch die zum Zeitpunkt des Mailversandes aktuellen Werte ersetzt.

### 5.5.3 SNMP Alarm

In Netzwerken mit einem Netzwerkmanagementsystem kann für die Übermittlung von Alarmen auch das SNMP Protokoll genutzt werden.

Es muss unter *Config >> Device >> Alarm >> Alarm1* zusätzlich zu den Alarmbedingungen *SNMP Trap Enable* aktiviert werden.

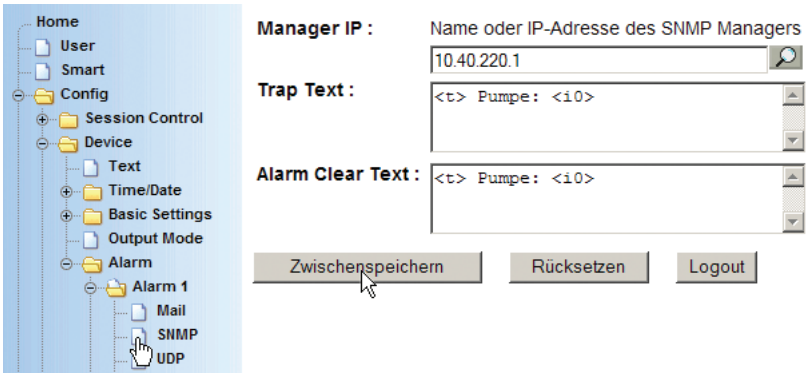
Notwendige Zugriffsrechte: *Administrator*




Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Wählen Sie im Navigationsbaum *Config* >> *Device* >> *Alarm* >> *Alarm1* >> SNMP

Notwendige Zugriffsrechte: *Administrator*




**Manager IP**
Geben Sie hier die IP-Adresse des Netzwerkmanagementsystems an, welches die SNMP-Informationen erhalten soll.

**Trap-Text**
Sie haben die Möglichkeit, genau wie bei der Mail-Funktion, auch hier einen „Trap Text“ einzugeben, der an die Manager-Software übertragen wird. Um den Status von Inputs und Outputs bzw. die Zählerstände der Counter in den Text einzubinden, können ebenfalls Alarm-Variablen genutzt werden.

**Alarm Clear Text**
Wie beim Trap Text kann hier ein frei formulierter Text für die Alarmclear-Meldung eingetragen werden.

*Die maximalen Längen von Trap-Text und Alarm Clear Text dürfen 450 Byte nicht überschreiten. Längere Texte werden vom Web-IO abgeschnitten.*

| Alarm Variable | Beschreibung |
|---|---|
| <dn> | Device Name (siehe: Config>>Device>>Text) |
| <i> | Zustand der Inputs als Bitmuster in hexadezimaler Schreibweise |
| <i**x**> | Zustand des Inputs Nr. **x** (ON/OFF), |
| <in**x**> | Name des Inputs Nr. **x** |
| <o> | Zustand der Outputs als Bitmuster in hexadezimaler Schreibweise |
| <o**x**> | Zustand des Outputs Nr. **x** (ON/OFF), |
| <on**x**> | Name des Outputs Nr.**x** |
| <c**x**> | Zählerstand des Counters Nr. **x** in dezimaler Schreibweise |
| <t> | Zeitstempel mit Datum und Uhrzeit im Format: TT.MMM.JJJJ hh:mm:ss |
| <$y> | Jahr im Format "JJJJ" |
| <$m> | Monat im Format "MM" |
| <$d> | Tag im Format "TT" |
| <$h> | Stunde im Format "hh" |
| <$i> | Minuten im Format "mm" |
| <$s> | Sekunden im Format "ss" |
| | **x** kann je nach Modell zwischen 0 und 23 liegen |

*Sollten Sie ein Administrator-Passwort eingegeben haben, müssen Sie dieses in Ihrer Manager-Software als „community string“ eintragen!*

### 5.5.4 Alarm über UDP

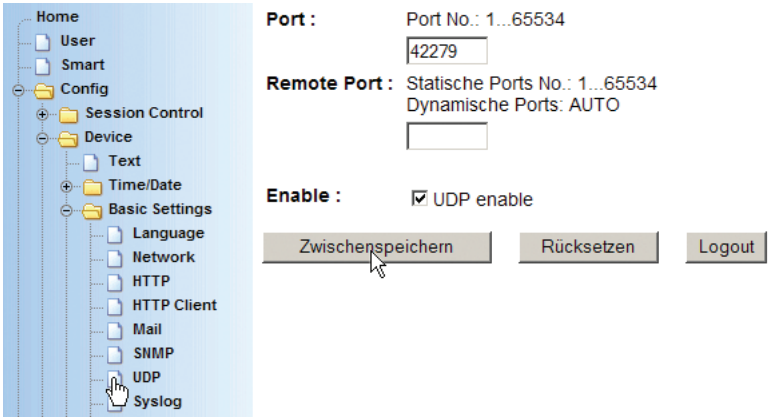
Eine weitere Möglichkeit, einen Alarm an einen anderen Netzwerkteilnehmer weiterzugeben, bietet das UDP-Protokoll. Hierzu muss auf den PCs, die einen solchen Alarm empfangen sollen, ein entsprechendes Anwendungsprogramm laufen.

Um UDP zur Alarmübertragung zu aktivieren, wählen Sie im Navigationsbaum *Config >> Basic Settings >> UDP*

Notwendige Zugriffsrechte: *Administrator*

Tragen Sie im Feld *Port* den gewünschten lokalen Port ein. Werksseitig ist Port 42279 voreingestellt. Das Feld *Remote Port* ist für das Versenden von Alarmen nicht relevant. Akti-

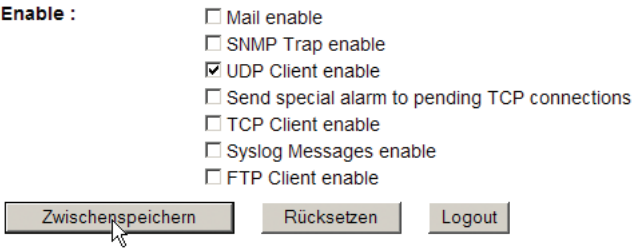
vieren Sie das Feld *UDP enable*.




Übergeben Sie die gewählten Einstellungen durch Klick auf den *Zwischenspeichern*-Button a das Web-IO.

Wählen Sie nun im Navigationsbaum *Config >> Device >> Alarm >> Alarm1* und markieren Sie im Bereich *Enable* den Punkt *UDP Client Enable.*

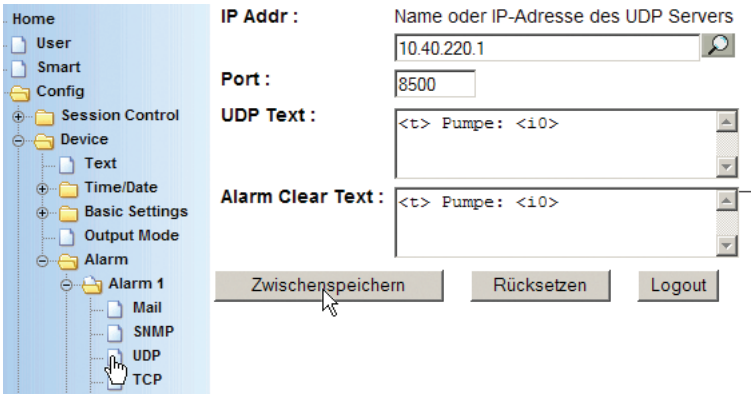
Notwendige Zugriffsrechte: *Administrator*




Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeicher*n-Button an das Web-IO.

Unter *Config >> Device >> Alarm >> Alarm1* >> UDP können dann die UDP-spezifischen Konfigurationen eingetragen werden.

Notwendige Zugriffsrechte: *Administrator*




**IP Addr**
Hier wird die IP-Adresse des Netzwerkteilnehmers eingetragen, der die Alarmmeldung empfangen soll. Alternativ kann eine Broadcast-IP-Adresse eingetragen werden, um die Alarmmeldungen allen Teilnehmern im Netzwerksegment zugänglich zu machen. Die IP-Adresse 10.255.255.255 z.B. würde allen Netzteilnehmern im Segment 10.0.0 erlauben, den Alarm zu empfangen.

**Port**
in diesem Feld wird der lokale UDP-Port eingetragen, auf dem das Anwendungsprogramm die UDP-Alarmmeldungen empfangen soll.

**UDP-Text**
Sie haben die Möglichkeit, genau wie bei der Mail- und SNMP-Funktion, hier einen Text einzugeben, der an die UDP-Anwendung übertragen wird. Um den Status von Inputs und Outputs, bzw. die Zählerstände der Counter in den Text einzubinden, können ebenfalls die E-Mail-Tags genutzt werden.

**Alarm Clear Text**
Wie beim UDP Text kann hier ein frei formulierter Text für die Alarmclear-Meldung eingetragen werden.

Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

**Outputs eines anderen Web-IO über UDP schalten**
Das Versenden von UDP-Datagrammen kann auch dazu genutzt werden, Kommandos zum Schalten von Outputs an ein weiteres Web-IO zu senden.

Dazu müssen als *IP Addr* die Adresse und als *Port* der Local Port des anderen Web-IO eingetragen werden.

*Es kann „localhost“ oder „127.0.0.1“ bzw. die eigene Adresse als Adresse eingegeben werden. Das Web-IO sendet das Kommando dann an sich selbst, was ein Schalten der eigenen Outputs über die Alarme erlaubt.*

Unter UDP Text wird das entsprechende Kommando zum Schalten eines Outputs eingegeben.

z.B

```
GET /    outputaccess0?PW=<password>&State=ON&NA=ON&
```

zum Einschalten von Output 0 bzw.

```
GET /    outputaccess0?PW=<password>&State=OFF&NA=ON&
```

zum Ausschalten.

Dem Kommando sollte durch Drücken der Enter-Taste ein Zeilenvorschub angefügt werden.

*Details zu den Kommandos finden Sie in Kapitel 5.6.2*

*Bitte beachten Sie, dass der Trigger für den Alarm min.eine Sekunde anstehen muss, damit das Kommando zuverlässig gesendet wird.*

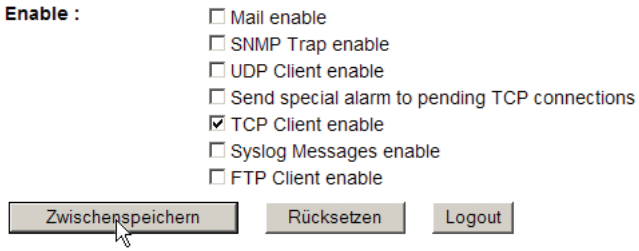
### 5.5.5 Alarmmeldung an eigene TCP-Applikationen

In Anwendungen, die über das Socket-Interface via TCP mit dem Web-IO kommunizieren, kann über bestehende TCP-Verbindungen bei Zustandsänderung der Input-Status übergeben

werden. Auf diese Weise lassen sich auch event-gesteuerte TCP-Programme erstellen.

Damit bei Eintreten der Alarmbedingung ein TCP-Datenpaket an die aktive TCP-Clientanwendung gesendet wird, markieren Sie unter *Config >> Device >> Alarm >> Alarm1* im Bereich *Enable* den Punkt *Send input to pending TCP connections.*

Notwendige Zugriffsrechte: *Administrator*




Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

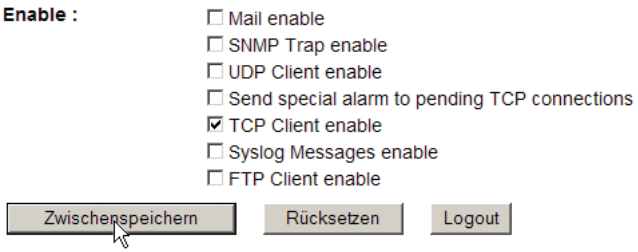
Wenn alle den Alarm betreffenden Eingaben gemacht wurden, aktivieren Sie die neuen Einstellungen über *Config >> Session Control >> LogOut, Speichern.*

### 5.5.6 Alarmmeldung an einen TCP-Server

Sollen Alarmmeldungen von einer TCP-Server Anwendung weiterverarbeitet werden, stellt das Web-IO eine TCP-Clientfunktion zur Verfügung, die im Alarmfall eine TCP-Verbindung zum gewünschten Server aufbaut und die Alarmmeldung überträgt. Der TCP-Server muss nach Empfang der Alarmdaten die Verbindung trennen!

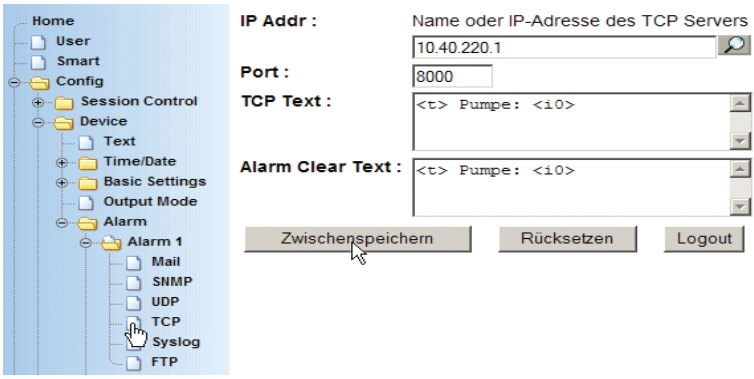
Um bei Eintreten der Alarmbedingung eine TCP-Verbindung aufzubauen und den Alarm zu versenden, markieren Sie unter *Config >> Device >> Alarm >> Alarm1* im Bereich *Enable* den Punkt *TCP Client enable.*

Notwendige Zugriffsrechte: *Administrator*




Übergeben Sie de Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Unter *Config >> Device >> Alarm >> Alarm1* >> TCP können dann die TCP-spezifischen Konfigurationen eingetragen werden.




**IP-Addr**
Hier wird die IP-Adresse des Netzwerkteilnehmers eingetragen, der die Alarmmeldung empfangen soll.

**Port**
in diesem Feld wird der lokale TCP-Port eingetragen, auf dem die Serveranwendung die TCP-Alarmmeldungen empfangen soll.

**TCP-Text**
Sie haben die Möglichkeit, genau wie bei der Mail- und SNMP-Funktion, hier einen Text einzugeben, der an die TCP-Serveranwendung übertragen wird. Um den Status von Inputs und Outputs, bzw. die Zählerstände der Counter in

den Text einzubinden, können ebenfalls die Alarm-Variablen genutzt werden.

| Alarm Variable | Beschreibung |
|---|---|
| <dn> | Device Name (siehe: Config>>Device>>Text) |
| <i> | Zustand der Inputs als Bitmuster in hexadezimaler Schreibweise |
| <i**x**> | Zustand des Inputs Nr. **x** (ON/OFF), |
| <in**x**> | Name des Inputs Nr. **x** |
| <o> | Zustand der Outputs als Bitmuster in hexadezimaler Schreibweise |
| <o**x**> | Zustand des Outputs Nr. **x** (ON/OFF), |
| <on**x**> | Name des Outputs Nr.**x** |
| <c**x**> | Zählerstand des Counters Nr. **x** in dezimaler Schreibweise |
| <t> | Zeitstempel mit Datum und Uhrzeit im Format: TT.MMM.JJJJ hh:mm:ss |
| <$y> | Jahr im Format "JJJJ" |
| <$m> | Monat im Format "MM" |
| <$d> | Tag im Format "TT" |
| <$h> | Stunde im Format "hh" |
| <$i> | Minuten im Format "mm" |
| <$s> | Sekunden im Format "ss" |
| | **x** kann je nach Modell zwischen 0 und 23 liegen |

**Alarm Clear Text**

Wie beim TCP Text kann hier ein frei formulierter Text für die Alarmclear-Meldung eingetragen werden.

Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Wenn alle den Alarm betreffenden Eingaben gemacht wurden, aktivieren Sie die neuen Einstellungen über *Config >> Session Control >> LogOut, Speichern.*

**Outputs eines anderen Web-IO über TCP schalten**

Das Versenden von TCP-Meldungen kann auch dazu genutzt werden, Kommandos zum Schalten von Outputs an ein weiteres Web-IO zu senden.

Dazu müssen als *IP Addr* die Adresse und als *Port* der Local Port (normal 80) des anderen Web-IO eingetragen werden.

*Es kann „localhost“ oder „127.0.0.1“ bzw. die eigene Adresse als Adresse eingegeben werden. Das Web-IO sendet das Kommando dann an sich selbst, was ein Schalten der eigenen Outputs über die Alarme erlaubt.*

Unter TCP Text wird das entsprechende Kommando zum Schalten eines Outputs eingegeben.

z.B

```
GET /    outputaccess0?PW=<password>&State=ON&NA=ON&
```

zum Einschalten von Output 0 bzw.

```
GET /    outputaccess0?PW=<password>&State=OFF&NA=ON&
```

zum Ausschalten.

Dem Kommando sollte durch Drücken der Enter-Taste ein Zeilenvorschub angefügt werden

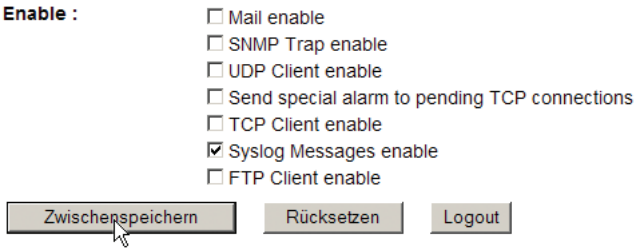
*Details zu den Kommandos finden Sie in Kapitel 5.6.2*

*Bitte beachten Sie, dass der Trigger für den Alarm min.eine Sekunde anstehen muss, damit das Kommando zuverlässig gesendet wird.*

### 5.5.7 Alarmmeldung an einen Syslog-Server

Um bei Eintreten der Alarmbedingung ein Systemmeldung an einen Syslog-Server zu senden, markieren Sie unter *Config >> Device >> Alarm >> Alarm1* im Bereich *Enable* den Punkt *Syslog Message enable.*

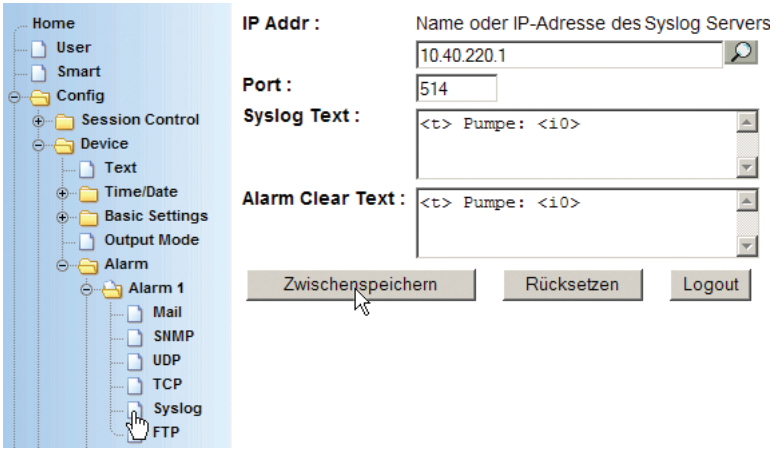
Notwendige Zugriffsrechte: *Administrator*




Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Unter *Config >> Device >> Alarm >> Alarm1* >> Syslog kön-

nen dann die Syslog-spezifischen Konfigurationen eingetragen werden.




**IP Addr**
Hier wird die IP-Adresse des Syslog-Servers eingetragen, der die Alarmmeldung empfangen soll.

**Port**
in diesem Feld wird der lokale UDP-Port eingetragen, auf dem der Syslog-Server die Syslog-Meldung empfangen soll.

Der Standardport für Syslog ist 514.

**Syslog-Text**
Sie haben die Möglichkeit, genau wie bei der Mail- und SNMP-Funktion, hier einen Text einzugeben, der an den Syslog-Server übertragen wird. Um den Status von Inputs und Outputs, bzw. die Zählerstände der Counter in den Text einzubinden, können ebenfalls die Alarm-Variablen genutzt werden.

| Alarm Variable | Beschreibung |
|---|---|
| <dn> | Device Name (siehe: Config>>Device>>Text) |
| <i> | Zustand der Inputs als Bitmuster in hexadezimaler Schreibweise |
| <i**x**> | Zustand des Inputs Nr. **x** (ON/OFF), |
| <in**x**> | Name des Inputs Nr. **x** |
| <o> | Zustand der Outputs als Bitmuster in hexadezimaler Schreibweise |
| <o**x**> | Zustand des Outputs Nr. **x** (ON/OFF), |
| <on**x**> | Name des Outputs Nr.**x** |
| <c**x**> | Zählerstand des Counters Nr. **x** in dezimaler Schreibweise |
| <t> | Zeitstempel mit Datum und Uhrzeit im Format: TT.MMM.JJJJ hh:mm:ss |
| <$y> | Jahr im Format "JJJJ" |
| <$m> | Monat im Format "MM" |
| <$d> | Tag im Format "TT" |
| <$h> | Stunde im Format "hh" |
| <$i> | Minuten im Format "mm" |
| <$s> | Sekunden im Format "ss" |
| | **x** kann je nach Modell zwischen 0 und 23 liegen |

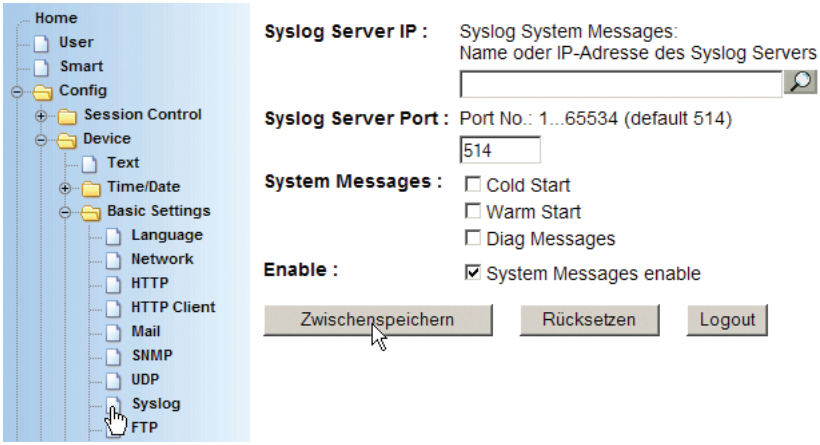
### Alarm Clear Text

Wie beim Syslog Text kann hier ein frei formulierter Text für die Alarmclear-Meldung eingetragen werden.

Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Zuletzt muss noch der Syslog Dienst freigeschaltet werden.

Wählen Sie hierzu im Navigationsbaum *Config >> Device >> Basic Settings >> Syslog.*



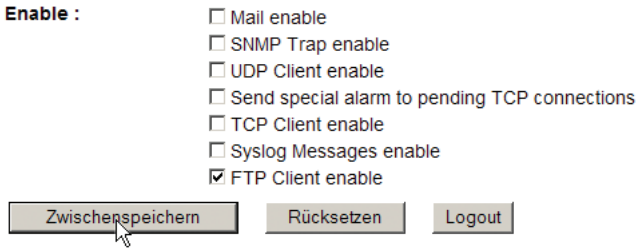

Durch Markieren von *System Messages enable* und Klick auf *Zwischenspeichern* wird Syslog freigegeben.

Wenn alle den Alarm betreffenden Eingaben gemacht wurden, aktivieren Sie die neuen Einstellungen über *Config >> Session Control >> LogOut, Speichern.*

### 5.5.8 Alarmmeldung an einen FTP-Server

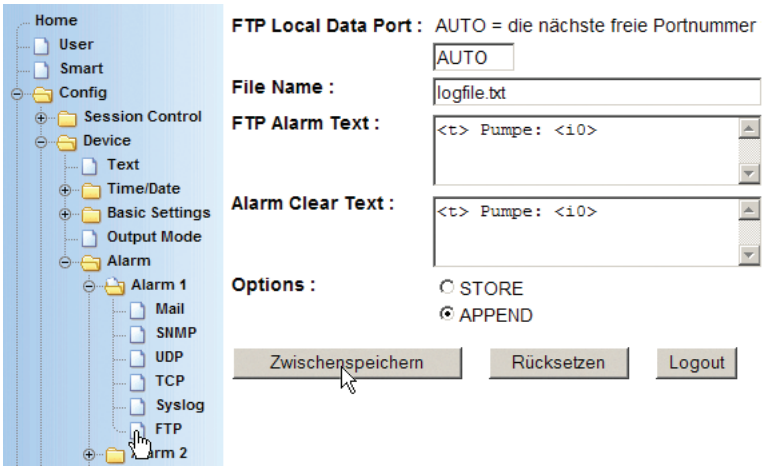
Um bei Eintreten der Alarmbedingung eine Meldung per FTP in eine Log-Datei auf einem FTP-Server zu schreiben, markieren Sie unter *Config >> Device >> Alarm >> Alarm1* im Bereich *Enable* den Punkt *FTP Client enable.*

Notwendige Zugriffsrechte: *Administrator*




Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Unter *Config >> Device >> Alarm >> Alarm1 >> FTP* können dann die FTP-spezifischen Konfigurationen eingetragen werden.




**FTP Local Data Port**
Hier wird der lokale Port der Verbindung angegeben, auf der das Web-IO die Alarmmeldungen an den FTP-Server versendet.

Wird AUTO angegeben, vergibt das Web-IO den nächsten freien Port.

**File Name**
in diesem Feld wird der Name der Datei eingegeben, in der die Meldungen abgespeichert werden sollen.

**FTP Alarm Text**
Sie haben die Möglichkeit, genau wie bei der Mail- und SNMP-Funktion, hier einen Text einzugeben, der an den Syslog-Server übertragen wird.

Zusätzlich können im File Namen und im FTP Alarm Text Variable eingesetzt werden. Das sind Platzhalter für Zustand und Namen der einzelnen Inputs und Outputs, sowie für die Zählerstände der Counter. Darüber hinaus können Platzhalter für Datum und Device Name eingesetzt werden.

| Alarm Variable | Beschreibung |
|---|---|
| <dn> | Device Name (siehe: Config>>Device>>Text) |
| <i> | Zustand der Inputs als Bitmuster in hexadezimaler Schreibweise |
| <i**x**> | Zustand des Inputs Nr. **x** (ON/OFF), |
| <in**x**> | Name des Inputs Nr. **x** |
| <o> | Zustand der Outputs als Bitmuster in hexadezimaler Schreibweise |
| <o**x**> | Zustand des Outputs Nr. **x** (ON/OFF), |
| <on**x**> | Name des Outputs Nr.**x** |
| <c**x**> | Zählerstand des Counters Nr. **x** in dezimaler Schreibweise |
| <t> | Zeitstempel mit Datum und Uhrzeit im Format: TT.MMM.JJJJ hh:mm:ss |
| <$y> | Jahr im Format "JJJJ" |
| <$m> | Monat im Format "MM" |
| <$d> | Tag im Format "TT" |
| <$h> | Stunde im Format "hh" |
| <$i> | Minuten im Format "mm" |
| <$s> | Sekunden im Format "ss" |
| | **x** kann je nach Modell zwischen 0 und 23 liegen |

### Alarm Clear Text

Wie beim FTP Alarm Text kann hier ein frei formulierter Text für die Alarmclear-Meldung eingetragen werden.

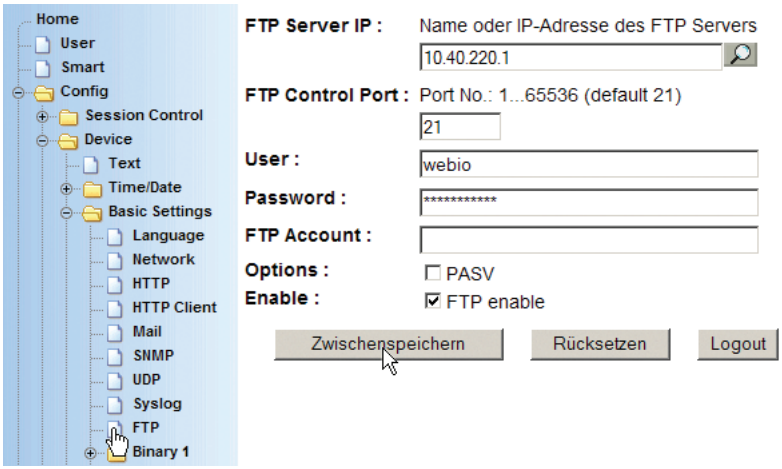
### Option

Wählen Sie *STORE*, wenn die Datei mit jeder eingehenden Meldung komplett überschrieben werden soll. Für eine mit jeder Meldung wachsenden Datei wählen Sie *APPEND*.

Übergeben Sie die Einstellungen durch Klick auf de *Zwischenspeichern*-Button an das Web-IO.

Zuletzt muss noch der FTP Dienst freigeschaltet und grundsätzlich konfiguriert werden.

Wählen Sie hierzu im Navigationsbaum *Config >> Device >> Basic Settings >> FTP.*




**FTP Server IP**
Geben Sie hier die IP-Adresse des FTP-Servers ein, auf dem die Alarmmeldungen gespeichert werden sollen.

**FTP Control Port**
An dieser Stelle wird der TCP-Port für die FTP Kontrollverbindung eingegeben. Im Normalfall arbeiten FTP-Server auf Port 21.

**User**
Hier wird der Name eines auf dem FTP-Server mit Schreibrechten registrierten Benutzers eingetragen.

**Password**
Zum Benutzer passendes, auf dem FTP-Server hinterlegtes Passwort

**FTP-Account**
Wenn vorhanden, wird hier das Konto des Benutzers eingetragen.

**Options**
Legen Sie hier fest, ob der FTP-Client aktiv oder passiv arbeiten soll. Fragen Sie im Zweifel den Administrator des FTP-Servers.

**Enable**
Durch Markieren von *Enable* und Klick auf *Zwischenspeichern* wird FTP als Dienst freigegeben.

Wenn alle den Alarm betreffenden Eingaben gemacht wurden, aktivieren Sie die neuen Einstellungen über *Config >> Session Control >> LogOut, Speichern.*

### 5.5.9 Allgemeine Informationen zu Alarmen

Für einen Alarm können mehrere und sogar alle Ausgabewege gleichzeitig gewählt werden. Es ist z.B. kein Problem, im Alarmfall eine E-Mail und einen SNMP-Trap zu versenden.

## 5.6 Zugriff aus der eigenen Anwendung

Um aus eigenen Anwendungen direkt auf das Web-IO zugreifen zu können, werden zwei grundsätzliche Wege angeboten:

- Zugriff aus dem Browser (HTTP-Request/Java-Applet)
- Zugriff aus diversen Hochsprachen (TCP/IP-Sockets)

### 5.6.1 Zugriff aus dem Browser

Von Hause aus erlaubt das Web-IO auf drei vorgegebenen Webseiten den Zugriff auf die Inputs, Outputs und Counter.

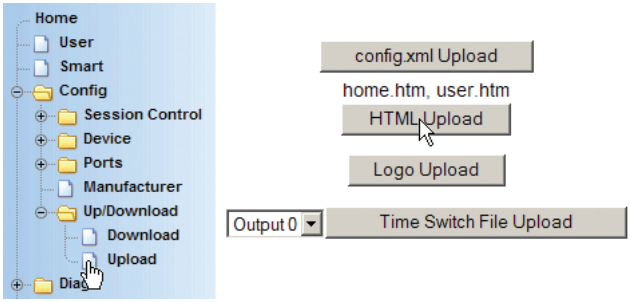
**home.htm**
- ✓ Dynamische Visualisierung der IOs und Alarme
- ✓ Schalten der Outputs nur nach Login über den Menübaum

**user.htm**
- ✓ Dynamische Visualisierung der IOs und der virtuellen Modbus-Register
- ✓ Schalten der Outputs nach speziellem Login auf der User-Seite möglich
- ✓ Das Passwort kann in einem Cookie gespeichert werden, so dass bei Aufruf vom selben PC nicht jedes mal ein Login erfolgen muss.

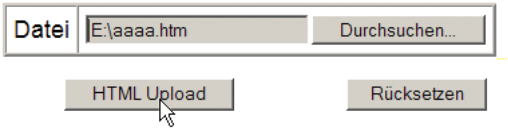
**smart.htm**
- ✓ Für Smartphones optimierte Webseite für die dynamische Visualisierung der IOs
- ✓ Schalten der Outputs nach speziellem Login möglich
- ✓ Das Passwort kann in einem Cookie gespeichert werden,- so dass bei Aufruf vom selben Smartphone nicht jedes mal ein Login erfolgen muss.

Alle drei Seiten können gegen selbst erstellte Webseiten ausgetauscht werden.

Wählen Sie dazu Im Menübaum *Config >> Up/Download >> Upload.*




und klicken Sie dann auf das HTML Upload Button.




Wählen Sie die hochzuladende Datei aus und bestätigen Sie mit Klick auf *HTML Upload*.

Der Name der Datei spielt keine Rolle. Ob die Datei als *home*, *user* oder *smart*-Seite eingespielt wird, ergibt sich über die erste Zeile des Seiten-Quelltextes.

```
<user.htm>
<!DOCTYPE html PUBLIC „-//W3C//DTD HTML 4.01 Transitional//EN“>
<html>
  <head>
    <meta........
    ..................
```

Das Web-IO entfernt die erste Zeile nach dem Hochladen, bevor die Seite gespeichert wird (deshalb gibt es auch kein zugehöriges schließendes Tag).

### Verwendung von W&T Tags

Wie in Kapitel „5.1.4 Beschriftung und Texte“ beschrieben, können das Web-IO selbst, Inputs, Outputs und Counter frei benannt und beschriftet werden.

Eine entsprechende Anzeige im Browser wird durch die Verwendung von W&T-Tags erreicht. W&T Tags sind Platzhalter,

die vom Web-IO bei der Übergabe der Webseite an den Browser durch die hinterlegten Bezeichnungen ersetzt werden. So kann erreicht werden, das ein und die gleiche Webseite in verschiedenen Web-IOs individuell anders aussieht.

Diese Tags setzen sich aus `<w&t_tags=` und dem eigentlichen Funktionsaufruf zusammen.

`<w&t_tags=time>`
zum Beispiel zeigt im Browser die aktuelle Systemzeit nebst Datum des Web-IO an.

Eine Besonderheit gibt es bei den W&T-Tags, die den Status der Outputs und die Zählerstände der Input-Counter wiedergeben.

`<w&t_tags=ox>` und `<w&t_tags=cx>`

Bei bestehendem Administrator- oder Operator-Login sind die angezeigten Inhalte (ON/OFF oder Zählerstand) mit einem Hyperlink versehen. Ein Klick auf diesen Link wechselt den Zustand des Outputs bzw. setzt den Counter auf 0.

Damit im Anschluss der geänderte Zustand angezeigt wird, lädt der Browser automatisch die Webseite neu.

Folgende W&T-Tags stehen zur Verfügung:

| Aufruf / Tag | Funktion |
|---|---|
| device_name | Fügt den für das Web-IO vergebenen Namen in die Webseite ein. |
| device_text | Fügt die für das Web-IO hinterlegte Beschreibung in die Webseite ein. |
| location | Fügt den für Location hinterlegten Eintrag in die Webseite ein |
| contact | Fügt den für Contact hinterlegten Eintrag in die Webseite ein |
| input**x** | Fügt die für Input Nr. **x** festgelegte Bezeichnung ein.<br>**x** kann sich dabei zwischen 0-11 bewegen und gibt an auf welchen Input sich der Aufruf bezieht.<br>Das gilt auch für die im Folgenden beschriebenen Output- und Counter-Aufrufe. |
| i**x** | Gibt den Zustand (ON/OFF) des **x** entsprechenden Inputs wieder. |
| bci**x** | Stellt den Hintergrund eines Tabellenfeldes abhängig vom Zustand des Input **x** dar.<br>OFF = keine Hintergrund Farbe, ON = grün. |
| c**x** | Fügt den Zählerstand des Counters für Input **x** in die Webseite ein. |
| output**x** | Fügt die für Output Nr. **x** festgelegte Bezeichnung ein. |
| o**x** | Gibt den Zustand (ON/OFF) des **x** entsprechenden Outputs wieder. Bei Login mit Config- oder Administratorrechten, ist die Zustandsangabe mit einem Hyperlink ausgestattet. Durch Mouse-Klick auf diesen Link wechselt der entsprechende Output seinen Zustand und die Webseite wird aktualisiert. |
| bco**x** | Stellt den Hintergrund eines Tabellenfeldes abhängig vom Zustand des Output **x** dar.<br>OFF = keine Hintergrund Farbe, ON = grün. |
| access:xxxx:yyyy | Kann an einem Link auf die Seiten home.htm, user.htm oder index.htm hinzugefügt werden. Bei Administrator-Login werden mehrere Outputs gleichzeitig geschaltet.<br>xxxx gibt in hexadezimaler Schreibweise an, welche Outputs sich ändern dürfen.<br>yyyy gibt in hexadezimaler Schreibweise an, in welchen Zustand die Outputs geschaltet werden sollen. |
| time | Fügt die Systemzeit und das Datum des Web-IO in die Webseite ein. |
| reload_button | Fügt einen Button mit der Funktion reload in die Webseite ein. |
| session | Bei Login mit Operator- oder Administratorrechten, generiert das Web-IO eine Session-ID. Bei erneutem Seitenaufruf muss diese Session-ID als Parameter angegeben werden, damit z.B. bei Rücksprung über einen Link, die Zugriffsrechte erhalten bleiben. |

### Ein Beispiel für die Verwendung von W&T-Tags

Der im folgende HTML-Quelltext zeigt den Umgang mit den W&T-Tags:

Da die W&T-Tages beim erst Laden in den Browser vom Web-IO gegen die tatsächlichen Inhalte ersetzt werden, muss die Webseite zunächst ins Web-IO gespeichert werden.

In diesem Fall als user-Seite.

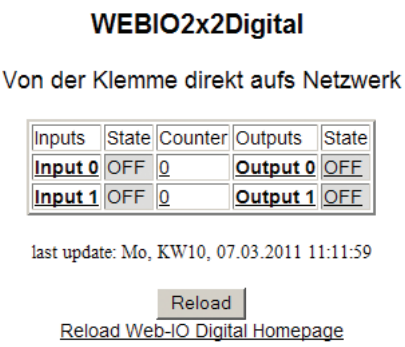
```
<user.htm>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <link rel="stylesheet" href="style.css" type="text/css">
```

Damit die Inhalte der W&T-Tags richtig angezeigt werden muss das Sylesheet *style.css* verlinkt werden

```
    <title>W&amp;T Web-IO Home</title>
  </head>
  <body>
    <div align="center">
      <p class="size5"><b><w&t_tags=device_name></b></p>
      <p class="size4"><w&t_tags=device_text></p>
      <table border="2">
        <tr>
          <td>Inputs</td>
          <td>State</td>
          <td>Counter</td>
          <td>Outputs</td>
          <td>State</td>
        </tr>
        <tr>
          <th><w&t_tags=input0></th>
          <td <w&t_tags=bci0>><w&t_tags=i0></td>
          <td><w&t_tags=c0></td>
          <th><w&t_tags=output0></th>
          <td <w&t_tags=bco0>><w&t_tags=o0></td>
        </tr>
        <tr>
          <th><w&t_tags=input1></th>
          <td <w&t_tags=bci1>><w&t_tags=i1></td>
          <td><w&t_tags=c1></td>
          <th><w&t_tags=output1></th>
          <td <w&t_tags=bco1>><w&t_tags=o1></td>
        </tr>
      </table>
      <p>last update: <w&t_tags=time></p>
```

```
      <form action=home.htm" method="GET" ><w&t_tags=reload_button></form><br>
      <a href="index.htm<w&t_tags=session>" target="_top">Reload Web-IO Digital
      Homepage</a>
    </div>
  </body>
</html>
```

Im Browser stellt sich die Seite so dar:




**Links und Rücklinks**

Einige Anwendungen machen es ggf. nötig, von der User- oder Home-Seite des Web-IO auf eine Webseite zu verlinken, die auf einem anderen HTTP-Server liegt.

Wird danach durch einen Rücklink wieder auf eine Seite des Web-IO zurücknavigiert, geschieht dies mit Standard-User Zugriffsrechten, auch wenn vorher ein Administrator- oder Operator-Login bestand.

Um einen Rücklink mit den bestehenden Zugriffsrechten zu ermöglichen, gibt es das <w&t_tags=session> Tag. Dieses Tag ist Platzhalter für eine Session-ID.

Der entsprechende Link müsste dann in die Seite eingefügt werden wie folgt:

```
<a  href="http://wut.hs2/webioback.htm<w&t_tags=session>"  target="_top">Testlink
</a>
```

Wird der Link aufgerufen, ersetzt das Web-IO das Session-Tag

durch die Session-ID und übergibt diese an die neuaufgerufene Webseite.

Um die Session-ID für einen Rücklink einzusetzen, muss in die Fremdseite ein kleines JavaScript eingefügt werden, welches die Session-ID bei Aufruf der Seite entgegennimmt und in den Rücklink einsetzt.

```
<html><head>

<script language="JavaScript">
session=window.location.search

function setbacklink() {
 var backlinkurl = "http://172.16.232.17/index.htm";
 var backlink = backlinkurl.concat(session);
 window.location.href = backlink;
}
</script>
```

Im Body-Bereich der Seite kann dann die Funktion *setbacklink* als Hyperlink eingesetzt werden.

```
</head>
  <body>
    Es wird nun mit der übergebenen Session eine Testseite aufgerufen.
    <a href="javascript:setbacklink();"  >Aufruf Testseite</a>
  </body>
</html>
```

So ist sichergestellt, dass nach Rücksprung zum Web-IO die Zugriffsberechtigung erhalten beibt.

**Dynamische Webseiten mit JavaScript und AJAX-Technik**

Der aktuelle Status von Inputs, Outputs und Countern kann über HTTP-Requests beim Web-IO abgefragt werden.

Folgende HTTP-Requests werden vom Web-IO unterstützt:

| HTTP-Request | Parameter | Beschreibung |
|---|---|---|
| input**x** | ?PW=**password**&<br>An Stelle von **password,** muss das Administrator- oder Configpasswort eingesetzt werden.<br>Wurde kein Passwort vergeben, wird "PW=&" angegeben.<br>***(gilt für alle Kommandos!)*** | Anforderung des Inputstatus<br>**x** kann ein Wert zwischen 0-11 sein und gibt den Input an.<br>Die Rückmeldung des Web-IO ist ein String beginnend mit "input**x;**" gefolgt vom Input-Status:<br>ON = Signal am Input und<br>OFF = kein Signal am Input<br>Wird **x** komplett weggelassen, gibt das Web-IO ein den Input-Signalen entsprechendes Bitmuster in hexadezimaler Schreibweise zurück. |
| counter**x** | ?PW=**password**& | Anforderung des Counterwertes<br>**x** kann ein Wert zwischen 0-11 sein und gibt den Input an.<br>Die Rückmeldung des Web-IO ist ein String beginnend mit "counter**x;**". Der Zählerstand des gewählten Counters wird in dezimaler Schreibweise angefügt. |
| counter | ?PW=**password**& | Anforderung aller Counter-Werte<br>Die Rückmeldung des Web-IO ist ein String beginnend mit "counter;". Die Zählerstände werden mit Semikolon getrennt in dezimaler Schreibweise angefügt. |
| output**x** | ?PW=**password**& | Anforderung des Outputstatus<br>**x** kann ein Wert zwischen 0-11 sein und gibt den Output an.<br>Die Rückmeldung des Web-IO ist ein String beginnend mit "output**x;**" gefolgt vom Output-Status:<br>ON = Signal am Output und<br>OFF = kein Signal am Output<br>Wird **x** komplett weggelassen, gibt das Web-IO ein den Output-Signalen entsprechendes Bitmuster in hexadezimaler Schreibweise zurück. |
| outputaccess**x** | ?PW=**password**&<br>[Mask=XXXX&]<br>State=**ON**/**OFF**/**YYYY**&<br>ON: Output = 1,<br>OFF: Output = 0,<br>TOGGLE: Output wechselt Zustand<br>XXXX: Hex-Wert zwischen 0000 und 0FFF entsprechend der Bits die gesetzt werden sollen<br>YYYY: Hex-Wert zwischen 0000 und 0FFF entsprechend dem Ausgangsbitmuster. | Setzen eines oder mehrerer Outputs<br>**x** kann ein Wert zwischen 0-11 sein und gibt den Output an, der gesetzt werden soll.<br>Die Rückmeldung des Web-IO ist ein String beginnend mit "output;" gefolgt von einem den Output-Signalen entsprechenden Bitmuster in hexadezimaler Schreibweise.<br>Die Angabe von Mask ist optional. Wird Mask nicht mitgesendet, gilt das outputaccess Kommando für alle Outputs. |
| counterclear**x** | ?PW=**password**&<br>[Set=**value**&]<br>value: Zählervoreinstellung, Wert zwischen<br>0 und 2 Milliarden | Setzt den Zählerstand eines Counters. Wird der Parameter Set nicht mitgesendet, ist die Voreinstellung 0.<br>**x** kann ein Wert zwischen 0-11 sein und gibt den Input an, dessen Counter zurückgesetzt werden soll.<br>Die Rückmeldung des Web-IO ist ein String beginnend mit "counter**x**;". Der neue Zählerstand des gewählten Counters wird in dezimaler Schreibweise angefügt.<br>Wird **x** nicht angegeben, werden alle Counter gesetzt. |
| allout | ?PW=**password**& | Sammelanforderung der Input, Output-Zustände und aller Counterwerte.<br>Das Web-IO antwortet mit einem String folgenden Aufbaus:<br>input;0xxx;output;0xxx;counter;n0;n1;n2,...<br>0xxx entspricht dabei dem Status der Inputs bzw. Outputs in hexadezimaler Schreibweise.<br>n0,n1 usw. enthalten die Zählestände in dezimaler Darstellung |

| HTTP-Request | Parameter | Beschreibung |
|---|---|---|
| time | ?PW=**password**& | Gibt die systemzeit des Web-IO im Format: DD.MM.YYYY,hh:mm:ss zurück. D=Tag, M=Monat, Y=Jahr, h=Stunde, m=Minute, s=Sekunde |
| settime | ?PW=**password**& time=**DD.MM.YYYY,hh:mm:ss**& | Setzt die Systemzeit des Web-IO auf den mit **time** übergebenen Wert |
| diagnosis | ?PW=**password**& | Fordert den Status des Diagnosespeichers an. Das Web-IO gibt zurück: diagnosis;0000;00000000;00000000,00000000 der vierstellige Wert gibt die Anzahl der gespeicherten Meldungen an. Bei den drei achtstelligen hexadezimalen Werten steht jedes gesetzte Bit stellvertretend für eine der 92 moglichen Meldungen. |
| diagnosis**x** | ?PW=**password**& | Mit **x** wird der Index für eine der aktuell gespeicherten Meldungen angegeben. Als Rückgabe sendet das Web-IO den entsprechenden Meldungstext. **x** darf nicht größer sein als die Anzahl der aktuell anliegenden Meldungen. |
| diaglist**x** | ?PW=**password**& | Gibt die Meldungen zu den einzelnen Meldungsbits zurück (max. 92) |
| diagclear | ?PW=**password**& | Löscht den Meldungsspeicher |
| errorclear | ?PW=**password**& | Löscht Load-Fehler und gibt die betroffenen Outputs wieder frei. Siehe Kapitel *Diagnose und Test* |
| modbusreg | ?PW=**password**& | Gibt bei aktivierter Modbus-Unterstützung die 128 Byte virtuellen Speicher zurück, die vom Modbus-Master frei beschrieben werden können: modbus;<1.Byte>;<2.Byte>;.....;<128.Byte> Es werden alle 128 Byte mit Semikolon getrennt als Hexadezimalzahl übergeben. |

### Ein Beispiel für die Verwendung von HTTP-Requests

Das folgende Beispiel zeigt, wie mit Hilfe von JavaScript und HTTP-Requests (AJAX-Technik) eine selbstaktualisierende Webseite für das Web-IO aufgebaut werden kann. Hier zunächst der reine HTML-Anteil der Webseite, der quasi als Anzeigegerüst für die AJAX-Technik dient:

```
<user.htm>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1250">
  <title>Web-IO Digital, User</title>
  <style type="text/css">
    * { font-family:arial; }
    table { font-size:14px; }
    .borderLeft { border-left:1px solid #000000; }
    .button { font-size:9px; width:40px; }
```

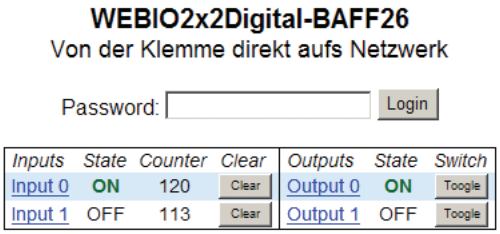
```
      .ce { text-align:center; }
      .description { font-size:18px; text-align:center; }
      .info { font-size:10px; text-align:center; }
      .italic { font-style:italic; }
      .name { font-size:20px; font-weight:bold; text-align:center }
      .sideSpace { padding-left:5px; padding-right:5px; }
      .table { background-color:#d6e8ff; border-collapse:collapse;
            border:1px solid #000000; }
      .whiteBack { background-color:#ffffff; }
   </style>
   <script language="JavaScript" type="text/javascript">
```

*hier steht eigentlich der JavaScript-Teil der Seite, der im Folgenden näher beschrieben wird.*

```
   </script>
   </head>
<body onload="CommandLoop();">
<div class="name"><w&t_tags=device_name></div>
<div class="description"><w&t_tags=device_text></div>
<br>
  <form>
    <div align="center">
              <span id="pt"">Password: </span>
         <input id="pw" type="password" name="tfPassword" maxlength="31"
size="20"">
              <input  id="lb" type="button" value="Login"
onclick="setPassword()">
        </div>
  </form>
<table align="center" class="table">
  <tr class="italic whiteBack">
    <td class="sideSpace">Inputs</td>
    <td class="sideSpace">State</td>
    <td class="ce sideSpace20">Counter</td>
    <td class="ce">Clear</td>
    <td class="borderLeft sideSpace">Outputs</td>
    <td class="sideSpace">State</td>
    <td class="ce">Switch</td>
  </tr>
  <tr>
    <td class="sideSpace"><w&t_tags=input0></td>
    <td class="ce" id="input0">-</td>
    <td class="ce" id="counter0">-</td>
```

```
    <td class="ce sidespace">
      <input class="button" onclick="clearCounter(0);" type="button"
value="Clear">
    </td>
    <td class="borderLeft sideSpace"><w&t_tags=output0></td>
    <td class="ce" id="output0">-</td>
    <td class="sideSpace">
      <input class="button" onclick="setOutput(0);" type="button" value="Toogle">
    </td>
  </tr>
  <tr class="whiteBack">
    <td class="sideSpace"><w&t_tags=input1></td>
    <td class="ce" id="input1">-</td>
    <td class="ce" id="counter1">-</td>
    <td class="ce sidespace">
      <input class="button" onclick="clearCounter(1);" type="button"
value="Clear">
    </td>
    <td class="borderLeft sideSpace"><w&t_tags=outpu1></td>
    <td class="ce" id="output1">-</td>
    <td class="sideSpace">
      <input class="button" onclick="setOutput(1);" type="button" value="Toogle">
    </td>
  </tr>
</table>
</body>
</html>
```

Im Browser sieht das so aus:




Die nachfolgend gezeigten Funktionen stützen sich auf zwei Grundtechniken:

- Identifikation und nachträgliche Veränderung eines HT-

ML-Objektes durch eindeutige ID
- Kommunikation mit dem Server (Web-IO) nach Laden der Web-Seite durch HTTP-Requests

Das zur Web-Seite gehörende JavaScript macht sich diese beiden Techniken zunutze.

Zunächst wird mit *maxi* und *maxo* festgelegt, wieviele Inputs und Outputs unterstützt werden sollen. Es wird eine Variable *applicationsstep* für die verschiedenen Programmschritte definiert. Als Abfrageintervall werden 500ms festgelegt und es wird eine Variable für das Passwort angelegt.

```
var maxi = 2;
var maxo = 2;
var applicationstep = 0;
var interval = 500;
var iopassword = ,';
```

Die Funktion *HextoInt* erlaubt die Umrechnung von hexadezimalen Zahlen in Ganzzahlenwerte (wird benötigt, wenn mit mehr als 2 IOs gearbeitet wird, da die Zustandsübergabe dann hexadezimal erfolgt).

```
function HexToInt(HexStr)
{ var TempVal;
  var HexVal=0;
  for( var i=0; i<HexStr.length;i++)
  { if (HexStr.charCodeAt(i) > 57)
    { TempVal = HexStr.charCodeAt(i) - 55;
    }
    else
    { TempVal = HexStr.charCodeAt(i) - 48;
    }
    HexVal=HexVal+TempVal*Math.pow(16, HexStr.length-i-1);
  }
  return HexVal;
}
```

Die Funktion *CommandLoop* setzt die HTTP-Requests zusammen und übergibt sie an die *DataRequest* Funktion.

```
function CommandLoop()
{ var commandstring ='';
  applicationstep++;
  switch(applicationstep)
  { case 1:
        commandstring = 'input?PW=' + iopassword + '&';
    break;
    case 2:
      commandstring = 'output?PW=' + iopassword + '&';
    break;
    case 3:
      commandstring = 'counter?PW=' + iopassword + '&';
      applicationstep = 0;
    break;
  }
  DataRequest(commandstring);
  maintimer = setTimeout("CommandLoop()", interval);
}
```

Die Funktion *DataRequest* ist das Herzstück dieses JavaScriptes. Sie nimmt den HTTP-Request entgegen und sendet ihn an den Server (das Web-IO). Die Funktion *DataRequest* nimmt aber auch die Antwort des Web-IO entgegen und übergibt sie an die Funktion *updateDisplay*.

```
  function DataRequest(SendString)
  { var xmlHttp;
    if( window.ActiveXObject )       // Internet Explorer
    { xmlHttp = new ActiveXObject( "Microsoft.XMLHTTP" );
    }
    else if(window.XMLHttpRequest )  // Mozilla, Opera und Safari
    { xmlHttp = new XMLHttpRequest();
    }
    if (xmlHttp)
    { xmlHttp.onreadystatechange = function()
      { if (xmlHttp.readyState == 4)
        { if (xmlHttp.status == 200)
          { if (xmlHttp.responseText.length > 0)
            { updateDisplay(xmlHttp.responseText);
            }
            xmlHttp=null;
```

```
            }
          }
        }
        xmlHttp.open("GET", SendString, true);
        xmlHttp.setRequestHeader("Connection", "close");
        xmlHttp.setRequestHeader("If-Modified-Since", "Thu, 1 Jan 1970 00:00:00
GMT");
        xmlHttp.send(null);
      }
    }
```

Die Funktion *updateDisplay* wertet die Antwort des Web-IO aus und passt die Anzeige im Browser entsprechend an. Dazu wird geprüft, ob sich die Antwort des Web-IO auf Inputs, Outputs oder Counter bezieht. Über *document.getElementById(ID)* identifiziert JavaScript die zu ändernden Objekte und passt deren Eigenschaften dem aktuellen IO-Status an.

```
function updateDisplay(ReceiveStr)
{  var HexVal;
   var state;
   var ReceiveData = ReceiveStr.split(';')
   // Display Intput state
   if (ReceiveData[ReceiveData.length - 2].substring(0, 1) == 'i')
   {  HexVal = HexToInt(ReceiveData[ReceiveData.length - 1]);
      for (var i = 0; i < 2; i++)
      {  state = false;
         if ((HexVal & Math.pow(2, i)) == Math.pow(2, i))
         {  state = true;
         }
         document.getElementById('input'+i).firstChild.data =( !state ) ?
'OFF':'ON';
         document.getElementById('input'+i).style.color
                                  = ( !state ) ? '#000000' : '#006600';
         document.getElementById('input'+i).style.fontWeight
                                    = ( !state ) ? 'normal' : 'bold';
      }
   }
   // Display Output state
   if (ReceiveData[ReceiveData.length - 2].substring(0, 1) == 'o')
   {  HexVal = HexToInt(ReceiveData[ReceiveData.length - 1]);
```

```
    for (var i = 0; i < 2; i++)
    { state = false;
      if ((HexVal & Math.pow(2, i)) == Math.pow(2, i))
      { state = true;
      }
      document.getElementById('output'+i).firstChild.data = ( !state ) ? 'OFF' :
'ON';
      document.getElementById('output'+i).style.color
                                = ( !state ) ? '#000000' : '#006600';
      document.getElementById('output'+i).style.fontWeight
                                  = ( !state ) ? 'normal' : 'bold';
    }
  }
  //Display Counter
  if (ReceiveData.length - maxi - 1 >= 0)
  { if (ReceiveData[ReceiveData.length - maxi - 1].substring(0, 1) == 'c')
    { for (var i = 0; i < maxi; i++)
      { document.getElementById('counter' + i).innerHTML
                          = ReceiveData[ReceiveData.length - maxi + i]
      }
    }
  }
  //Display cleared Counter
  if (ReceiveData[ReceiveData.length - 2].substring(0, 1) == 'c')
  { document.getElementById('counter'
  + ReceiveData[ReceiveData.length - 2].substring(7, ReceiveData[ReceiveData.
length -      2].length)).innerHTML = ReceiveData[ReceiveData.length - 1];
  }
}
```

Die Funktion *setOutput* sendet das entsprechende Kommando zum toggeln des gewählten Output an die Funktion *DataRequest*.

```
function setOutput(iNr)
{ var commandstring = 'outputaccess'+iNr+'?PW='+iopassword+'&State=TOGGLE&';
  DataRequest(commandstring);
}
```

Die Funktion *clearCounter* sendet das entsprechende Kommando zum Löschen des Zählerstandes des gewählten Input

an die Funktion *DataRequest*.

```
function clearCounter(iNr)
{  DataRequest('counterclear'+iNr+'?PW='+iopassword+'&');
}
```

Die Funktion *setPassword* übernimmt das eingegebene Passwort und schreibt es in die Variable *iopassword*, die Bestandteil der Kommandostrings ist.

```
function setPassword()
{  iopassword = document.getElementById('pw').value;
   document.getElementById('pw').value = '';
}
```

*Die auf HTTP-Requests gestützte AJAX-Technik kann nur eingesetzt werden, wenn die Webseite aus dem Web-IO geladen wird, das auch die Requests empfangen soll.*

Die hier gezeigten HTTP-Requests lassen sich gleichermaßen mit anderen Web-Techniken wie z.B. PHP nutzen.

**Nutzen des Web-IO eigenen Java-Applets**

Im Gegensatz zur AJAX-Technik können die IO-Zustände des Web-IOs mit Hilfe des Java-Applets auch in Webseiten dynamisch angezeigt werden, die von einem Dritt-Server geladen werden.

Die Einbindung des Applets erfolgt über das <applet> HTML-Tag, wobei die Standardparameter *archiv* und *code* gesetzt werden wie folgt:

```
archive="dio.jar"
code="dio.class"
```

Die weiteren Parameter werden individuell passend zur Anwendung gesetzt

```
name="dio0"
```

Der Bezeichner *dio* wurde hier willkürlich gewählt und kann gegen jeden beliebigen anderen ersetzt werden. Es ist nur

wichtig, dass beim Aufruf der Applet-Funktionen durch JavaScript der gleiche Name verwendet wird.

```
codebase="http://<IP-Adresse >"
```

Wird die Webseite, aus der das Applet geladen wird als *user.htm* oder *home.htm* direkt aus dem Web-IO geladen, kann auf den Parameter *codebase* verzichtet werden.

```
width="0"     Angabe in Pixel
height="0"    Angabe in Pixel
```

Da das Applet keine visuelle Komponente enthält, sollten *width* und *hight* immer gleich 0 sein.

```
mayscript
```

Diese Option erlaubt es dem Java-Applet, auf JavaScript-Code im HTML-Dokument zuzugreifen.

Darüber hinaus können noch einige weitere Web-IO spezifische Parameter gesetzt werden, die jeweils in einem eigenen <param> Tag eingefügt werden:

**device**
Ganzzahliger Wert zwischen 0 und 255. Der *device* Parameter dient zur Indizierung, wenn auf einer Webseite mehr als ein Web-IO bedient werden soll.

Wird mit nur einem Web-IO gearbeitet, setzen Sie den Wert auf 0. Bei allen weiteren Web-IO, die in die Webseite eingebunden werden, zählt der Wert um 1 hoch.

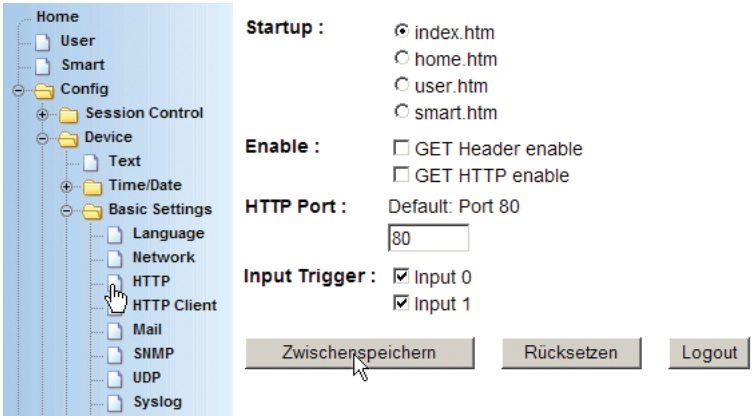
**showerrors**
Treten bei der Verwendung des Java-Applets Fehler auf, kann über diesen Parameter festgelegt werden, ob eine Fehlermeldung in Form eines Alert-Fensters ausgegeben werden soll.

**inputpolling**
Mit Hilfe dieses Parameters wird beim Starten des Applets festgelegt, ob die Inputs im unter *pollingrate* eingestellten In-

tervall auf ihren Zustand überprüft werden sollen.

Als *value* kann für *inputpolling on* oder *off* gesetzt werden.

Um die Datenlast auf dem Netzwerk zu reduzieren, kann über den Navigationsbaum des Web-IO unter *Config >> Device >> Basic Settings >> HTTP* der *Input Trigger* für die zu überwachenden Inputs gesetzt werden.




In diesem Fall kann *value* auf *off* gesetzt werden. Es entsteht für die Aktualisierung der Inputs dann nur noch Datenlast, wenn sich ein Input ändert.

Damit die Änderungen des Input Triggers wirksam werden, müssen die neuen Einstellungen unter Session *Control >> Logout* gespeichert werden.

**outputpolling**
Mit Hilfe dieses Parameters wird beim Starten des Applets festgelegt, ob die Outputs im unter *pollingrate* eingestellten Intervall auf ihren Zustand überprüft werden sollen.

Als *value* kann für *outputpolling on* oder *off* gesetzt werden.

**counterpolling**
Mit Hilfe dieses Parameters wird beim Starten des Applets festgelegt, ob die Counter im unter *pollingrate* eingestellten Intervall auf ihren Zustand überprüft werden sollen.

Als *value* kann für *outputpolling* auf *on* oder *off* gesetzt werden.

**pollingrate**
Mit diesem Parameter wird durch *value* festgelegt, in welchem Intervall (in ms) Inputs, Outputs oder Counter aktualisiert werden sollen.

Hier ein Beispiel:

```
<applet name="dio0" archive="dio.jar" code="dio.class"
      codebase="http://172.16.232.11" width="0" height="0" mayscript>
  <param name="device" value="0">
  <param name="showerrors" value="off">
  <param name="inputpolling" value="on">
  <param name="outputpolling" value="on">
  <param name="counterpolling" value="on">
  <param name="pollingrate" value="2000">
</applet>
```

*Sowohl bei den Parametern, als auch bei den Werten,ist auf Groß- und Kleinschreibung zu achten.*

**Die Verwendung des Applets mit JavaScript**
Um mit dem Applet arbeiten zu können, müssen im Kopf der Web-Seite entsprechende JavaScript-Funktionen deklariert werden.

Für den rein lesenden Zugriff sind das 4 Funktionen:

**Passwortschutz**

```
<script language="JavaScript" type="text/javascript">
<!--
function setPassword()
  {  document.applets[„dio0“].setPassword(‚Administratorpasswort‘);}
```

Wurde für das Web-IO ein Administrator- oder Operator-Passwort gesetzt, muss die Funktion *setPassword* aufgerufen werden, damit das Applet auf das Web-IO zugreifen kann. Anstatt eines statischen Passwortes kann das Passwort natürlich auch über ein Eingabefeld vom User abgefragt und überge-

ben werden.

In jedem Fall darf setPassword erst aufgerufen werden, wenn das Java-Applet gestartet wurde, da es sonst zu Laufzeitfehlern kommt.

*Ein Beispiel in dem das Passwort über ein Eingabefeldübergeben wird, finden Sie, wenn Sie im Web-IO die Webseite app.htm aufrufen.*

### Lesende Funktionen

```
function inputChanged( iDevice, iNr, iVal )
  {  Programmcode der beim Wechsel an den Inputs ausgeführt wird}
function outputChanged( iDevice, iNr, iVal )
  {  Programmcode der beim Wechsel an den Outputs ausgeführt wird}
function counterChanged( iDevice, iNr, iVal )
  {  Programmcode der bei Änderung eines den Counters ausgeführt wird}
```

Die drei vorstehenden Funktionen werden vom Applet aufgerufen, wenn eine Änderung an den Inputs bzw. den Outputs oder den Countern erkannt wird. iDevice gibt an, bei welchen Web-IO sich ein Wert geändert hat. Mit *iNr* wird übergeben, welcher Input, Output oder Counter sich geändert hat. Die Variable *iVal* übergibt den aktuellen Zustand.

*Bitte beachten Sie, dass bei den Namen der Funktionen zwischen Groß- und Kleinschreibung unterschieden wird.*

Der folgende Quelltext zeigt ein kleines Beispiel für die dynamische Anzeige von Input 0.

```
<html>
  <head>
    <script language="JavaScript" type="text/javascript">
    <!--
    function inputChanged( iDevice, iNr, iVal )
    {  if (iNr == 0 && iDevice == 0)
      {  document.getElementById( 'input' + iNr ).firstChild.nodeValue
          = ( !iVal ) ? 'OFF' : 'ON';
      }
```

```
      else
      {   iVal = 0;
      }
    }
    function setPassword()
    {  document.applets["dio0"].setPassword( 'wut');
    }
    //-->
    </script>
  </head>
  <body>
    <applet name="dio0" archive="dio.jar" code="dio.class"
      codebase="http://172.16.232.11" width="0" height="0">
     <param name="device" value="0">
     <param name="showerrors" value="off">
     <param name="inputpolling" value="on">
      <param name="outputpolling" value="off">
      <param name="counterpolling" value="off">
      <param name="pollingrate" value="1500">
    </applet>
    <p>State of Input 0:</p>
    <p id="input0">OFF</p>
    <script language="JavaScript" type="text/javascript">
    <!--
      setPassword();
    //-->
    </script>
  </body>
</html>
```

Das Beispiel beinhaltet nur die Funktion *inputChanged*, da im Bereich des Applet-Tags nur das *inputpolling* aktiviert wurde.

⚠ *Für die Verarbeitung der Inputs, Outputs und Counter gilt: Ist das entsprechende Polling aktiviert, muss die dazugehörige JavaScript Funktion im HTML-Text deklariert sein, damit es nicht zu Laufzeitfehlern kommt.*

⚠ *Soll das Passwort bei Aufruf der Web-Seite automatisch übergeben werden, muss der Aufruf der setPassword Funktion hinter dem Applet-Tag stehen.*

Das Passwort ist in diesem Fall unverschlüsselt lesbar im Quelltext hinterlegt. Aus Sicherheitsgründen empfehlen wir deshalb die Passwortabfrage über ein Eingabefeld.

**Schreibende Funktionen**

Um auch einen individuellen Zugriff auf die Outputs zu erlauben, stellt das Applet entsprechende Funktionen zur Verfügung.

```
function setOutput( iDevice, iOutput, iValue )
     {  iOut = 0;
        iOut |= Math.pow( 2, iOutput );
        document.applets["dio"+iDevice].outputAccess( iOut, iValue );
     }
```

Wenn Sie per JavaScript auf die Outputs zugreifen möchten, sollte vorstehende Funktion im Kopf der Webseite eingefügt werden. *iDevice* gibt an, bei welchem Web-IO ein Output geschaltet werden soll. Über *iOutput* wird übergeben, welcher Output gesetzt werden soll, *iValue* wird auf *0x0FFF* gesetzt wenn der Output auf ON schalten soll. *0x000* entspricht OFF.

Innerhalb der Funktion wird das JavaScript-Kommando *document.applets[„dio“*+iDevice*].outputAccess( iOut, iValue )* verwendet. Wenn Sie mehr als einen Output gleichzeitig schalten möchten, können Sie dieses Kommando in eigenen Funktionen verwenden. Die Variable *iOut* ist ein ganzzahliger 16Bitwert. Durch z.B. mathematische Funktionen oder Übergabe von Konstanten können die Bits, die verändert werden sollen, auf 1 gesetzt werden. In *iValue* wird für die einzelnen Bits angegeben, ob der entsprechende Output 0 (=OFF) oder 1 (=ON) sein soll.

Beispiel in hexadezimaler Schreibweise:

iOut = 0x0013, iValue = 0x0101

In diesem Fall werden die Outputs 0 und 4 auf ON und der Output 1 auf 0 gesetzt. Alle anderen Outputs bleiben unverändert.

Eine weitere Funktion ist für das Löschen der Counter vorhanden

```
function clearCounter( iDevice, iValue )
     {  iVal = 0;
        iVal |= Math.pow( 2, iValue );
        document.applets["dio"+iDevice].counterClear( iVal );
     }
```

Beim Aufruf dieser Funktion wird über *iValue* übergeben, welcher der Counter gelöscht werden soll.

*Ein umfassendes Beispiel für die Nutzung des Java-Applets zeigt die Webseite app.htm, die im Web-IO aufgerufen werden kann.*

Aufruf: http://<ip-adresse>/app.htm

### 5.6.2 Zugriff aus diversen Hochsprachen

Für den Zugriff über TCP/IP-Sockets bietet das Web-IO zwei Zugänge.

Zugriff über:

- Kommandostrings ASCII
- Binärstrukturen BINARY

**Kommandostrings ASCII**
Je nach Konfiguration arbeitet das Web-IO in diesem Modus als TCP-Server, als TCP-Client oder als UDP-Peer.

Durch den Austausch einfacher Kommandostrings können die Inputs und Counter gelesen bzw. die Outputs gesetzt werden.

Das Lesen der Inputs geschieht im Regelfall im Pollingverfahren. Eine ereignisgesteuerte Auswertung ist nur nach entsprechender Konfiguration möglich.

Vorteile:

- Da alle Hochsprachen wie z.B. Visual Basic, Visual Basic for Applications, Delphi, C, C++ ...usw. einen unkomplizierten Umgang mit Stringvariablen erlauben, kommen auch ungeübte Programmierer sehr schnell zum gewünschten Ergebnis.
- Unter TCP wird der Port 80 (sonst HTTP) benutzt, der in den meisten Firewalls nicht gesperrt ist. Somit ist in aller Regel kein Eingriff in die Netzwerkinfrastruktur nötig.
- Die Kombination aus der Benutzung von Stringvariablen und TCP-Port 80 macht eine Integration in PHP und CGI-Scripte sehr einfach. Auch aus SPS und anderen Mikrocontrollern kann das WEB-IO angesprochen werden, wenn eine Netzwerkanbindung vorhanden ist.

Nachteil:

- Nur bedingt ereignisgesteuerte Programmierung möglich. Dadurch für zeitkritische Anwendungen nur eingeschränkt nutzbar.

Fazit: Der Zugriff über Kommandostrings erlaubt es in kürzester Entwicklungszeit Anwendungen zu erstellen, bei denen die Verarbeitungsgeschwindigkeit nicht die oberste Priorität hat.

**Binärstrukturen BINARY**
In diesem Modus kann das Web-IO als TCP-Client, TCP-Server oder UDP-Peer arbeiten. Der Zugriff kann über ein Passwort geschützt werden.

Für die verschiedenen Funktionen wie Lesen der Inputs, Setzen der Outputs usw. definiert das Web-IO binäre Strukturen. Der Zugriff erfolgt ausschließlich durch Austausch dieser Strukturen.

Vorteile:

- Schnelle Kommunikation durch Verwendung von Strukturen. Die benötigten Statuswerte stehen an definierter

Stelle und es müssen keine Strings auf ihren Inhalt untersucht werden.
- Durch ereignisgesteuerte Kommunikation kann die Anwendung auf Änderungen an den Inputs sofort reagieren
- Hohe Anwendungssicherheit durch passwortgeschützten Zugriff möglich

Nachteil:

- In Hochsprachen die keine Binärstrukturen unterstützen, muss ein Workaround in Form von Binärstrings oder Byte-Arrays geschaffen werden.

Fazit: Für professionelle Anwendungen die eine hohe Verarbeitungsgeschwindigkeit benötigen, sollte der BINARY Modus zum Einsatz kommen.

**Socketprogrammierung mit Kommandostrings**
Um das Web-IO Digital aus einfachen Anwendungsprogrammen anzusprechen, ist ein direkter Zugriff über TCP oder UDP-Sockets möglich.

Der Datenaustausch zwischen PC und Web-IO geschieht dabei durch die Übergabe von einfachen Kommandostrings.

Die Kommandostrings bestehen aus:

der einleitenden Sequenz: `GET /`
dem eigentlichen Kommando: `<command>`
dem Trenner: `?`
und einem oder mehreren Parametern: `<parameter1>&<...`

Alle Kommandostrings enden immer mit `&`

Beispiel:

```
GET /input&PW=mypassword&
```

Bei einem mit dem Passwort *mypassword* geschützten Web-IO wird so der Zustand der Inputs abgefragt.

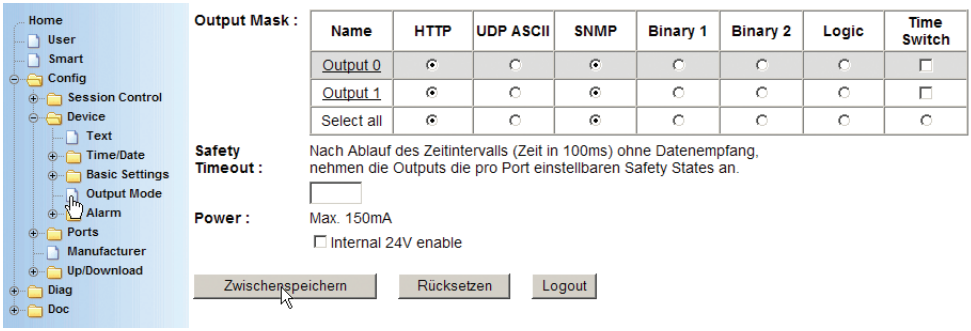
| Kommandos | Parameter | Beschreibung |
|---|---|---|
| GET /input**x** | ?PW=**password**&<br>An Stelle von **password,** muss das Administrator- oder Configpasswort eingesetzt werden.<br>Wurde kein Passwort vergeben, wird "PW=&" angegeben.<br>***(gilt für alle Kommandos!)*** | Anforderung des Inputstatus<br>**x** kann ein Wert zwischen 0-11 sein und gibt den Input an.<br>Die Rückmeldung des Web-IO ist ein String beginnend mit "input**x;**" gefolgt vom Input-Status:<br>ON = Signal am Input und<br>OFF = kein Signal am Input<br>Wird **x** komplett weggelassen, gibt das Web-IO ein den Input-Signalen entsprechendes Bitmuster in hexadezimaler Schreibweise zurück. |
| GET /counter**x** | ?PW=**password**& | Anforderung des Counterwertes<br>**x** kann ein Wert zwischen 0-11 sein und gibt den Input an.<br>Die Rückmeldung des Web-IO ist ein String beginnend mit "counter**x;**". Der Zählerstand des gewählten Counters wird in dezimaler Schreibweise angefügt. |
| GET /counter | ?PW=**password**& | Anforderung aller Counter-Werte<br>Die Rückmeldung des Web-IO ist ein String beginnend mit "counter;". Die Zählerstände werden mit Semikolon getrennt in dezimaler Schreibweise angefügt. |
| GET /output**x** | ?PW=**password**& | Anforderung des Outputstatus<br>**x** kann ein Wert zwischen 0-11 sein und gibt den Output an.<br>Die Rückmeldung des Web-IO ist ein String beginnend mit "output**x;**" gefolgt vom Output-Status:<br>ON = Signal am Output und<br>OFF = kein Signal am Output<br>Wird **x** komplett weggelassen, gibt das Web-IO ein den Output-Signalen entsprechendes Bitmuster in hexadezimaler Schreibweise zurück. |
| GET /outputaccess**x** | ?PW=**password**&<br>[Mask=XXXX&]<br>State=**ON**/**OFF**/**YYYY**&<br>[NA=ON&]<br>ON: Output = 1,<br>OFF: Output = 0,<br>TOGGLE: Output wechselt Zustand<br>XXXX: Hex-Wert zwischen 0000 und 0FFF entsprechend der Bits die gesetzt werden sollen<br>YYYY: Hex-Wert zwischen 0000 und 0FFF entsprechend dem Ausgangsbitmuster. | Setzen eines oder mehrerer Outputs<br>**x** kann ein Wert zwischen 0-11 sein und gibt den Output an, der gesetzt werden soll.<br>Die Rückmeldung des Web-IO ist ein String beginnend mit "output;" gefolgt von einem den Output-Signalen entsprechenden Bitmuster in hexadezimaler Schreibweise.<br>Die Angabe von Mask ist optional. Wird Mask nicht mitgesendet, gilt das outputaccess Kommando für alle Outputs.<br>Mit NA=ON wird optional erreicht, dass vom Web-IO keine Antwort auf das outputaccess Kommando gesendet wird. |
| GET /counterclear**x** | ?PW=**password**&<br>[Set=**value**&]<br>value:<br>Zählervoreinstellung,<br>Wert zwischen<br>0 und 2 Milliarden | Setzt den Zählerstand eines Counters. Wird der Parameter Set nicht mitgesendet, ist die Voreinstellung 0.<br>**x** kann ein Wert zwischen 0-11 sein und gibt den Input an, dessen Counter zurückgesetzt werden soll.<br>Die Rückmeldung des Web-IO ist ein String beginnend mit "counter**x;**". Der neue Zählerstand des gewählten Counters wird in dezimaler Schreibweise angefügt.<br>Wird **x** nicht angegeben, werden alle Counter gesetzt. |
| GET /allout | ?PW=**password**& | Sammelanforderung der Input, Output-Zustände und aller Counterwerte.<br>Das Web-IO antwortet mit einem String folgenden Aufbaus:<br>input;0xxx;output;0xxx;counter;n0;n1;n2,...<br>0xxx entspricht dabei dem Status der Inputs bzw. Outputs in hexadezimaler Schreibweise.<br>n0,n1 usw. enthalten die Zählestände in dezimaler Darstellung |

| Kommandos | Parameter | Beschreibung |
|---|---|---|
| GET /time | ?PW=**password**& | Gibt die systemzeit des Web-IO im Format: DD.MM.YYYY,hh:mm:ss zurück. D=Tag, M=Monat, Y=Jahr, h=Stunde, m=Minute, s=Sekunde |
| GET /settime | ?PW=**password**& time=**DD.MM.YYYY,hh:mm:ss**& | Setzt die Systemzeit des Web-IO auf den mit **time** übergebenen Wert |
| GET /diagnosis | ?PW=**password**& | Fordert den Status des Diagnosespeichers an. Das Web-IO gibt zurück: diagnosis;0000;00000000;00000000,00000000 der vierstellige Wert gibt die Anzahl der gespeicherten Meldungen an. Bei den drei achtstelligen hexadezimalen Werten steht jedes gesetzte Bit stellvertretend für eine der 92 moglichen Meldungen. |
| GET /diagnosis**x** | ?PW=**password**& | Mit **x** wird der Index für eine der aktuell gespeicherten Meldungen angegeben. Als Rückgabe sendet das Web-IO den entsprechenden Meldungstext. **x** darf nicht größer sein als die Anzahl der aktuell anliegenden Meldungen. |
| GET /diaglist**x** | ?PW=**password**& | Gibt die Meldungen zu den einzelnen Meldungsbits zurück (max. 92) |
| GET /diagclear | ?PW=**password**& | Löscht den Meldungsspeicher |
| GET /errorclear | ?PW=**password**& | Löscht Load-Fehler und gibt die betroffenen Outputs wieder frei. Siehe Kapitel *Diagnose und Test* |

## TCP Kommunikation - das Web-IO als Server

Vorbereitend muss festgelegt werden, welche der Outputs für den Zugriff über Kommandostrings zur Verfügung stehen sollen.

Wählen Sie im Navigationsbaum *Config >> Device >> Output Mode*




Notwendige Zugriffsrechte: *Administrator*

Alle verwendeten Outputs müssen für *HTTP* aktiviert sein (Werkseinstellung).

Sollten Sie die Output-Modi verändert haben, klicken Sie auf den *Logout-Button. Durch* Klick auf den *Speichern-Button* werden die Einstellungen übernommen.

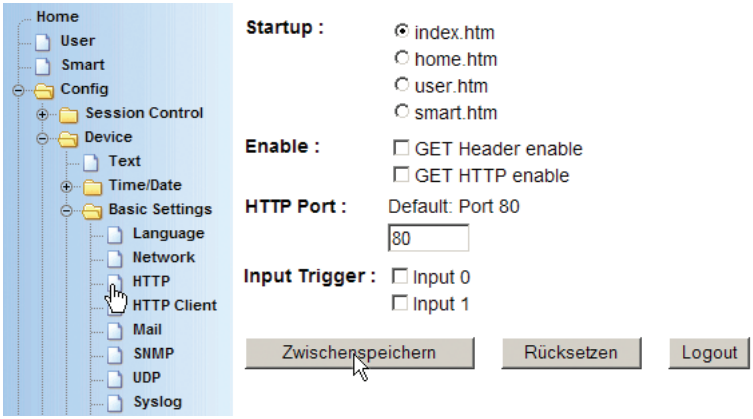
Das Web-IO arbeitet nun als TCP-Server, woraus sich unabhängig von der verwendeten Programmiersprache eine Dreiteilung des Programmablaufes ergibt.

**1. Aufbau der TCP-Verbindung**
Die Anwendung arbeitet in jedem Fall als Client und baut somit die TCP-Verbindung auf.

Das Socket-Interface des Web-IO ist in weiten Teilen an das HTTP-Protokoll angelehnt. Daraus ergibt sich, dass der Listenport ab Wer auf TCP-Port 80 eingestellt ist bzw. dem konfigurierten HTTP Port entspricht.

Soll ein abweichender Port benutzt werden, ist es möglich, den HTTP Port unter Config >> Device >> Basic Settings >> HTTP zu ändern.

Notwendige Zugriffsrechte: *Administrator*




Unter HTTP Port kann hier ein vom Standard Port 80 abweichender TCP Port für den HTTP-Zugriff eingetragen werden.

*Wird der HTTP Port geändert, ist das Web-IO auch aus dem Browser nur noch über die zusätzliche Angabe des*

*gewählten HTTP Ports erreichbar.*

Beispiel für HTTP Port = 85.

*Das gilt auch für alle Konfigurationszugriffe.*

⚠ *Zum Schutz vor blockierenden TCP-Verbindungen hat das Web-IO auf dem HTTP Port einen Timer eingesetzt, der die Verbindung 30 Sekunden nach Verbindungsaufbau automatisch wieder trennt, wenn keine syntaktisch korrekten Kommandos empfangen wurden. Nach erfolgreichem Verbindungsaufbau sollte die Anwendung also immer z.B. eine Abfrage der Outputs an das Web-IO senden.*

**2. Kommunikation zwischen Web-IO und Anwendung**

Die normale Kommunikation findet im Pollingbetrieb statt. Das bedeutet: die Client-Anwendung fordert mit Hilfe der Kommandostrings die gewünschten Werte an bzw. setzt die Outputs.

Beispiel 1: Abfrage des Status von Input 1, an dem ein Signal von +24V anliegt. Es wurde kein Administratorpasswort für das Web-IO vergeben.

Die Anwendung sendet folgenden String zum Web-IO:

```
GET  /input1?PW=&
```

Das Web-IO sendet daraufhin zurück:

```
input1;ON
```

Beispiel 2: Setzen der Outputs 0 und 1. Binär ergibt das 0011, was wiederum 3 hexadezimal ergibt.

Das Administratorpasswort für das Web-IO lautet: „blau“

Die Anwendung sendet folgenden String zum Web-IO:

```
GET /outputaccess?PW=blau&State=3&
```

Das Web-IO sendet daraufhin zurück:

```
output;3
```

*Bei Web-IO Boxen mit mehr als 2 Outputs werden die Werte vierstellig hexadezimal übergeben (output;0003)*

*Das Web-IO arbeitet mit null-terminierten Strings! Das bedeutet: alles was das Web-IO zur Anwendung sendet, endet mit einem 0-Character.*

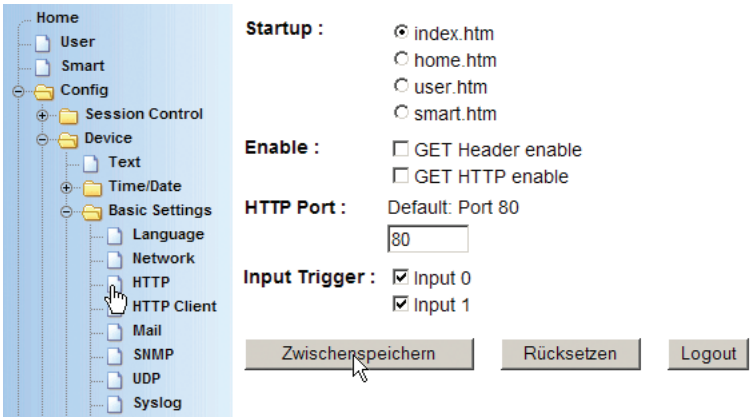
Bei Web-IO mit Relaisausgängen, muss zwischen zwei Schaltvorgängen an einem Output ein Pause von min. 200ms eingehalten werden!

**Event-gesteuerte Benachrichtigung**
Einige Anwendungen machen es notwendig, dass Änderungen an bestimmten Inputs automatisch über die bestehende TCP-Verbindung gemeldet werden.

Ein Grund kann die Reduzierung der Datenlast auf dem Netzwerk sein. Werden die Inputs in einem zyklischen Intervall abgefragt, erzeugt das eine kontinuierliche Datenlast. Ferner besteht die Gefahr, dass Änderungen an den Inputs während zweier Zyklen nicht erkannt werden.

Um die Bedingungen für eine Event-gesteuerte Benachrichtigung festzulegen, wählen Sie im Navigationsbaum *Config >> Device >> Basic Settings >> HTTP*

Notwendige Zugriffsrechte: *Administrator*




**Input Trigger**

Legen Sie hier fest, welche Inputs auf Zustandsänderung überwacht werden sollen. Bei bestehender Verbindung sendet das Web-IO automatisch den neuen Zustand der Inputs an die Anwendung.

**Einen Header mitsenden lassen**

Das Web-IO kann bei Bedarf dem gesendet String einen Header voranstellen, der die IP-Adresse und den Namen enthält.

Beispiel:

- Die IP-Adresse des Web-IO ist 10.40.22.126.
- Als Device Name wurde unter *Config >> Device >> Text* „W&T Web-IO 2x Digital Input, 2x Digital Output“ eingegeben.
- Mit *GET /input?PW=&* wurden von der Anwendung die Inputs abgefragt.

Das Web-IO gibt zurück:

*10.40.22.126;W&T Web-IO 2x Digital Input, 2x Digital Output;input;0000*

Wurde *GET Header enable* aktiviert, wird der Header allen Datensendungen vom Web-IO vorangestellt.

Wenn alle Bedingungen festgelegt wurden, aktivieren Sie die

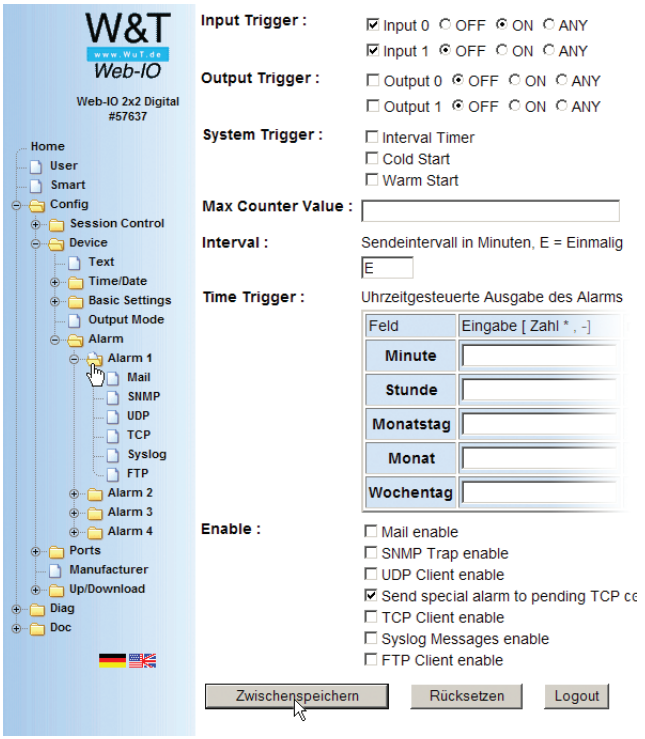
neuen Einstellungen über *Config >> Session Control >> LogOut.*

**Benachrichtigung bei definierten Inputmustern**
Bei einigen Anwendungen ist es sinnvoll, den Zustand der Inputs nur bei bestimmten Inputmustern zu erhalten.

*Die hier beschriebene Art der Benachrichtigung kann zwar parallel zum normalen Event-gesteuerten Betrieb verwendet werden - das bringt aber in der Praxis keine Vorteile, da in diesem Fall ja ohnehin bei jeder Änderung eine Benachrichtigung erfolgt. Beim Auftreten des festgelegten Inputmusters würde also doppelt benachrichtigt.*

Um die Bedingungen für eine TCP-Alarmierung bei bestimmten Inputmustern festzulegen, wählen Sie im Navigationsbaum *Config >> Device >> Alarm >> Alarm 1.*

Notwendige Zugriffsrechte: *Administrator*




**Input Trigger**

Legen Sie hier fest, welche Inputs auf welchen Zustand überwacht werden sollen. Wenn mehrere Inputs in einem Alarm zur Überwachung aktiviert werden, besteht zwischen den Bedingungen eine UND-Verknüpfung.

Beispiel:

In Alarm 1 wurde Input 0 auf ON aktiviert, Input 1 auf OFF.

Ein Alarm wird nur dann an die TCP-Anwendung hochgereicht, wenn beide Bedingungen erfüllt werden. Also Input 0= ON, Input 1 = OFF.

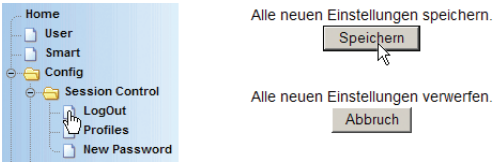
**Interval**

Für die Alarmierung über eine bestehende TCP-Verbindung ist es sinnvoll, hier den Parameter *E* einzusetzen, damit nur einmalig bei Eintreten der Alarmbedingung alarmiert wird.

**Enable**
Aktivieren Sie im Bereich *Enable* das Feld *Send input to pending TCP connections.*

Übergeben Sie die gewählten Alarmbedingungen jeweils durch Klick auf den *Zwischenspeichern*- Button an das Web-IO.

Wenn alle Alarmbedingungen festgelegt wurden, aktivieren Sie die neuen Einstellungen über *Config >> Session Control >> LogOut*




Nach einem Klick auf den *Speichern*-Button wird das Web-IO nun mit den aktuellen Parametern neu gestartet.

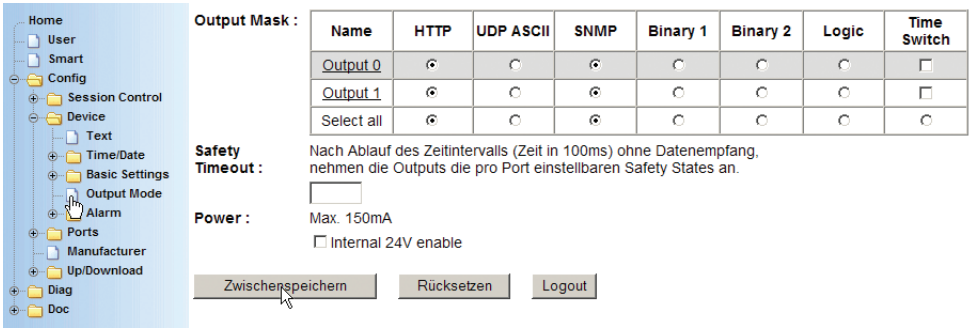
**3. Schließen der TCP-Verbindung**
Hier weicht das Web-IO in seinem Verhalten vom HTTP-Protokoll ab. Im Gegensatz zum HTTP-Protokoll ist es immer die Client-Anwendung, die eine Verbindung schließt.

*Ausnahme: Bei Empfang eines fehlerhaften Kommandostrings beendet das Web-IO sofort die Verbindung.*

**TCP Kommunikation - das Web-IO als Client**
In einigen Anwendungsfällen ist es nötig, dass das Web-IO als Client arbeitet. In diesem Fall baut das Web-IO eine TCP-Verbindung zu einer Server-Applikation auf. Auslöser für den Verbindungsaufbau kann eine Statusänderung an einem der Inputs sein oder es kann ein zeitgesteuerter Verbindungsaufbau konfiguriert werden.

Vorbereitend muss festgelegt werden, welche der Outputs, nach Verbindungsaufbau für den Zugriff über Kommando-Strings zur Verfügung stehen sollen.

Wählen Sie im Navigationsbaum *Config >> Device >> Output Mode*




Notwendige Zugriffsrechte: *Administrator*

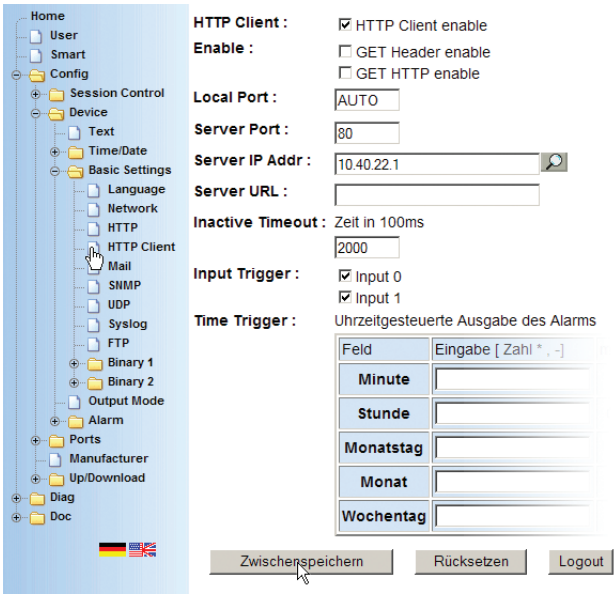
Mindestens einer bzw. alle verwendeten Outputs müssen für *HTTP* aktiviert sein.

Sollten Sie die Output-Modi verändert habe, klicken Sie auf den *Zwischenspeichern*-Button.

Abschließend muss das Web-IO für die TCP-Client-Kommunikation konfiguriert und aktiviert werden.

Wählen Sie im Navigationsbaum *Config >> Basic Settings >> HTTP-Client.*

Notwendige Zugriffsrechte: *Administrator*




**HTTP Client**

Dieses Feld muss aktiviert werden, damit das Web-IO als TCP-Client arbeitet.

**Enable**

Aktivieren Sie dieses Feld, wenn den Daten, die das Web-IO an die Server-Anwendung sendet, ein Header vorangestellt werden soll.

Beispiel:

- Die IP-Adresse des Web-IO ist 10.40.22.126.
- Als *Device Name* wurde unter *Config >> Device >> Text* „W&T Web-IO 2x Digital Input, 2x Digital Output“ eingegeben.
- Mit *GET /input?PW=&* wurden von der Server-Anwendung die Inputs abgefragt.

Das Web-IO sendet:

*10.40.22.126;W&T Web-IO 2x Digital Input, 2x Digital Output;input;0000*

Wurde *GET Header enable* aktiviert, wird der Header allen Datensendungen vom Web-IO vorangestellt. Bei Server-Anwendungen, die mehr als eine Verbindung annehmen können, kann so sehr einfach unterschieden werden, von wo die Datensendung kommt.

**Local Port**
Tragen Sie im Feld *Local Port* den gewünschten lokalen Port ein. Werksseitig ist *AUTO* voreingestellt und das Web-IO verwendet den nächsten freien Port.

**Server Port**
An dieser Stelle wird der Port eingetragen, auf dem die Server-Anwendung die Verbindung entgegen nimmt.

**Server IP-Adress**
Hier wird die IP-Adresse des PC eingegeben, zu dem verbunden werden soll.

**Server URL**
URL-Adresse des Servers - nur für Web-Anwendungen relevant

**Inactivity Timeout**
Nach der hier eingestellten Timeout Zeit, beendet das Web-IO automatisch die Verbindung zum Server. Die Timeout Zeit startet, sobald kein Datenverkehr mehr auf der Verbindung stattfindet.

**InputTrigger**
Hier müssen alle Inputs markiert werden, die bei Statusänderung einen Verbindungsaufbau bzw. ein Input-Status Paket zur Server-Anwendung auslösen sollen.

**Time Trigger**
In diesem Bereich kann ein zeitgesteuerter Verbindungsaufbau konfiguriert werden.

| Feld | Eingabe [ Zahl * , -] | möglicher Zahlenbereich |
|---|---|---|
| **Minute** | 0 | 0-59 |
| **Stunde** | 0-23/2 | 0-23 (0 ist Mitternacht) |
| **Monatstag** | * | 1-31 |
| **Monat** | * | 1-12 |
| **Wochentag** | 1-5 | 0-6 (0 ist Sonntag) |

Die Syntax der Zeiteinstellung ist an das in Linux/Unix verwendete Cron-Format angelehnt. In einem Feld können mehrere Angaben mit Komma getrennt eingegeben werden.

Das „*“ steht für jeden zulässigen ganzzahligen Wert. Beispiel: * im Feld Monat gilt für jeden Monat.

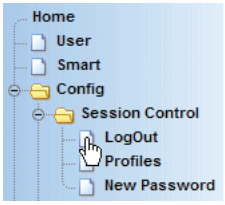
Mit „-“ können Zeitspannen als Trigger bestimmt werden. Beispiel: 1-5 im Feld Wochentag bedeutet Mo - Fr

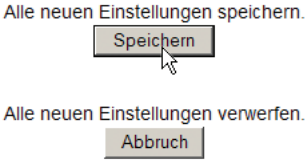
Wird dem eigentlichen Wert oder Wertebereich ein „/“ nachgestellt, kann dahinter eine Zykluswert angegeben werden. Beispiel: 0-23/2 im Feld Stunde ergibt alle zwei Stunden einen Alarm, wenn auch die Bedingungen in den anderen Feldern erfüllt sind. Wildcards („*“) sind in Verbindung mit dem „/“ nicht erlaubt (*/2 würde kein gesichertes Ergebnis bringen).

Die oben ausgefüllten Felder bedeuten also: jeden Monat, an jedem Montag bis Freitag, alle vollen zwei Stunden.

Übergeben Sie die gewählten Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Wenn alle Einstellungen festgelegt wurden, aktivieren Sie diese über *Config >> Session Control >> LogOut*




Nach einem Klick auf den *Speichern*-Button arbeitet das Web-IO als TCP-Client.

**Kommunikation zwischen Web-IO und Anwendung**

**1. Aufbau der TCP-Verbindung**
Die TCP-Verbindung wird nach den konfigurierten Triggerbedingungen vom Web-IO aufgebaut.

Nach erfolgreichem Verbindungsaufbau sendet das Web-IO den Input-Status im Format

```
input;0000
```

an die Server-Anwendung.

Der vierstellige Zahlenwert (beim 2fach Web-IO eine Stelle) gibt den Input-Status in hexadezimaler Schreibweise wieder.

*Das Web-IO arbeitet mit Null-terminierten Strings! Das bedeutet: alles was das Web-IO zur Anwendung sendet, endet mit einem 0-Character.*

**2. Kommunikation zwischen Web-IO und Anwendung**
Bei bestehender Verbindung kann die Server-Anwendung, genau wie eine Client-Anwendung über Kommandostrings mit dem Web-IO kommunizieren.

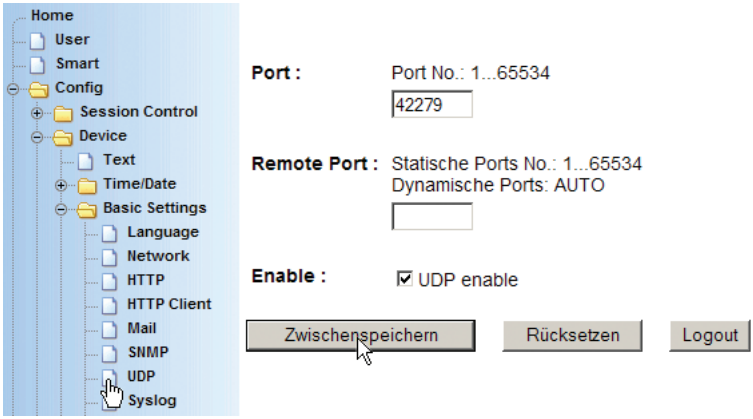
**3. Abbau der TCP-Verbindung**
Der Verbindungsabbau wird vom Web-IO automatisch eingeleitet, wenn nach dem letzten Datenverkehr zwischen Web-IO und Server-Anwendung die eingestellte Timeout Zeit verstrichen ist.

Natürlich kann auch die Server-Anwendung zu jedem beliebigen Zeitpunkt die Verbindung beenden.

**UDP-Kommunikation**
Zunächst muss das Web-IO für die Kommunikation via UDP-Kommandostrings aktiviert und der lokale Port eingetragen werden.

Wählen Sie im Navigationsbaum *Config >> Basic Settings >> UDP*

Notwendige Zugriffsrechte: *Administrator*




**Port**
Tragen Sie im Feld *Port* den gewünschten lokalen Port ein auf dem UDP Kommandos empfangen werden sollen. Werksseitig ist Port 42279 voreingestellt.

**Remote Port**
Tragen Sie im Feld *Remote Port* den Port ein, an den Datagramme vom Web-IO gesendet werden sollen. Werksseitig ist das Feld leer, was bedeutet, dass Antwort-Datagramme vom Web-IO an den Port zurückgehen, von dem die Kommandos auf der Gegenseite gesendet wurden.

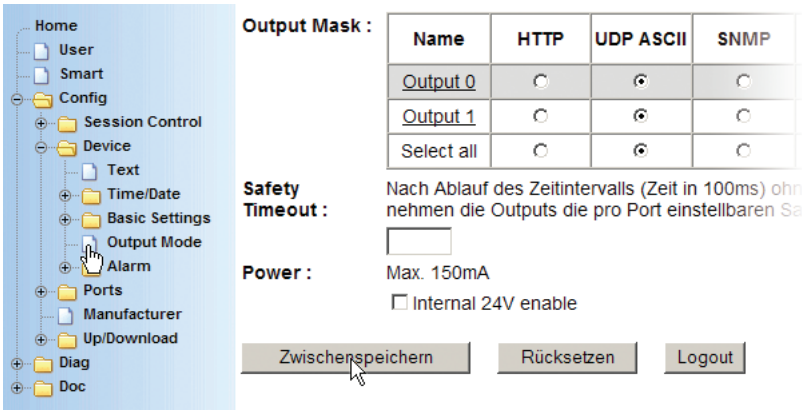
**Enable**
Aktivieren Sie das Feld UDP-enable.

Übergeben Sie die gewählten Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Ferner muss festgelegt werden, welche der Outputs für den Zugriff über Kommandostrings via UDP zur Verfügung stehen sollen.

Wählen Sie im Navigationsbaum *Config >> Device >> Output Mode.*

Notwendige Zugriffsrechte: *Administrator*




Alle verwendeten Outputs müssen für *UDP ASCII* aktiviert sein.

Sollten Sie die Output Modi verändert haben, klicken Sie auf den *Zwischenspeichern-Button* und anschließend auf den Link *alle zwischengespeicherten Einstellungen speichern und aktivieren*. Durch Klick auf den *Speichern-Button* werden die Einstellungen aktiviert.

**Kommunikation zwischen Web-IO und Anwendung**

Im Gegensatz zu TCP, gibt es bei der Kommunikation über das UDP-Protokoll keine feste Verbindung zwischen dem Web-IO und dem anderen Kommunikationspartner. Der Austausch von Daten findet über Datagramme statt.

Die normale Kommunikation findet im Pollingbetrieb statt.

Das bedeutet, die Anwendung fordert mit Hilfe der Kommandostrings die gewünschten Werte an, bzw. setzt die Outputs.

Die Antwort auf ein Polling sendet das Web-IO immer an die IP-Adresse des anfragenden Netzwerkteilnehmers zurück.

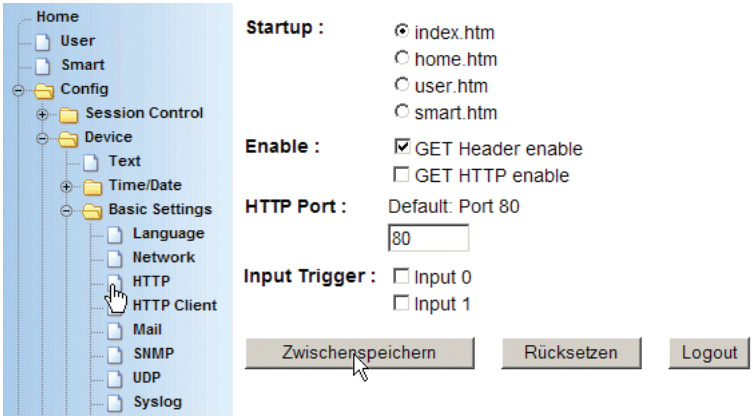
Die Kommunikation erfolgt bis auf den Verbindungsauf- und abbau identisch mit der bei TCP.

Um unter UDP bei Zustandsänderungen automatisch eine Meldung zu bekommen, stellt das Web-IO eine Fülle verschiedener Alarmbedingungen zur Verfügung.

*Eine detaillierte Beschreibung zur Nutzung dieser Alarmmöglichkeiten finden Sie im Kapitel* Alarme.

**Einen Header mitsenden lassen**

Das Web-IO kann bei Bedarf dem gesendet String einen Header voranstellen, der die IP-Adresse und den Namen enthält.




Um den Header freizuschalten, wählen Sie im Navigationsbaum *Config >> Basic Settings >> HTTP* und markieren Sie *GET Header enable*. Durch Klick auf *Logout* und *Speichern* wird das Versenden des Headers aktiviert.

Beispiel:

- Die IP-Adresse des Web-IO ist 10.40.22.126.
- Als Device Name wurde unter *Config >> Device >> Text* „W&T Web-IO 2x Digital Input, 2x Digital Output“ eingegeben.
- Mit *GET /input?PW=&* wurden von der Anwendung die Inputs abgefragt.

Das Web-IO gibt zurück:

*10.40.22.126;W&T Web-IO 2x Digital Input, 2x Digital Output;input;0*

Wurde *GET Header enable* aktiviert, wird der Header allen Datensendungen vom Web-IO vorangestellt.

**Programmgesteuertes Fehlermanagement**
Erkennt das Web-IO bei der Abwicklung seiner Aufgaben einen oder mehrere Fehler, werden diese gespeichert und sind über den Navigationsbaum unter *Diag* einsehbar.

Eine nähere Beschreibung hierzu finden Sie im Kapitel 6.6 Diagnose und Test.

In einigen Anwendungen ist es gewünscht, das Fehlermanagement nicht manuell durch Ansehen einer Webseite abzuwickeln, sondern Fehler automatisiert in einem Programm auszuwerten.

Hierzu stellt das Web-IO einige Kommandostrings zur Verfügung, mit denen gleichermaßen per TCP oder UDP gearbeitet werden kann.

**Erstellen einer Liste möglicher Fehlermeldungen**
Das Web-IO kann maximal 92 verschiedene Fehlertypen verwalten. Die tatsächliche Anzahl kann innerhalb dieser Grenze je nach Firmware-Stand variieren.

Um einen Überblick über die möglichen Fehler zu bekommen, kann das Kommando

```
GET /diaglistx?PW=password&
```

an das Web-IO gesendet werden.

Für *x* wird ein Wert zwischen 0 und 92 eingesetzt. Anstelle von *password* setzen Sie das Administrator- oder Operator-Passwort ein.

Das Web-IO beantwortet die Anfrage mit der entsprechenden Fehlermeldung.

```
diaglistx;errortext
```

Beispiel:

Sendet die Anwendung zu Web-IO (Passwort = „wut“)

```
GET /diaglist2?PW=wut&
```

antwortet das Web-IO mit

```
diaglist2;Formatfehler in der DNS Anfrage
```

Mit einer *for next* Schleife können über das *diaglist* Kommando alle möglichen Fehlermeldungen ausgelesen und als statische Liste abgespeichert werden. Im Binary-Modus zum Beispiel kann nur ermittelt werden wieviele Fehler aktuell vorliegen. Zusätzlich werden die zugehörigen Fehlernummern übergeben. Mit Hilfe der statischen Fehlerliste kann dann ein lesbarer Fehler ausgegeben werden.

### Direktes Auswerten der aktuellen Fehler

Um den aktuellen Fehlerstatus des Web-IO zu bekommen, wird das Kommando

```
GET /diagnosis?PW=password&
```

verwendet.

Das Web-IO gibt daraufhin zurück

```
diagnosis;iiii;yyyyyyyy;xxxxxxxx;zzzzzzzz
```

mit *iiii* gibt das Web-IO die Anzahl der aktuell vorliegenden Fehler in vierstelliger, hexadezimaler Schreibweise zurück.

*yyyyyyyy; xxxxxxxx,zzzzzzzz* können für normale Standardanwendungen außer Acht gelassen werden.

Für den Programmierprofi: *es handelt sich bei* yyyyyyyy; xxxxxxxx,zzzzzzzz *um drei 32Bit Werte in jeweils achtstelliger hexadezimaler Schreibweise. Die damit wiedergegebenen 92 Bit stehen stellvertretend für die 92 möglichen Fehlertypen. Mit Hilfe der statischen Fehlerliste (siehe GET /diaglist...) können die einzelnen Bits dekodiert werden. Dabei wird das niederwertigste Bit rechts im x Bereich angezeigt und das höchstwertige links im y Bereich.*

Um die aktuellen Fehler als Fehlertext zu bekommen, stellt das Web-IO das Kommando

```
GET /diagnosis x? PW=&
```

zur Verfügung.

x ist hierbei der Index für den Fehler in dezimaler Schreibweise beginnend bei 1.

Die Antwort vom Web-IO sieht dann so aus

```
diagnosis x;errortext
```

Im Gegensatz zu den Fehlertexten die mit *diaglist* aus dem Web-IO gelesen werden, enthält der Fehlertext bei *diagnosis* ggf. auch dynamische Elemente wie z.B. IP-Adressen oder Portnummern.

Ein Beispiel:

Die Anwendung sendet:

```
GET /diagnosis?PW=wut&
```

Das Web-IO antwortet:

```
diagnosis;0005;00400040;008A0000;00000000
```

Die Anwendung sendet:

```
GET /diagnosis1?PW=wut&
```

Das Web-IO antwortet:

```
diagnosis1;Mail Server antwortet nicht.
```

Die Anwendung sendet:

```
GET /diagnosis2?PW=wut&
```

Das Web-IO antwortet:

```
diagnosis2;Das Versenden der Mail wurde abgebrochen und wird wiederholt.
```

Die Anwendung sendet:

```
GET /diagnosis3?PW=wut&
```

Das Web-IO antwortet:

```
diagnosis3;Ziel IP-Adresse unbekannt: 172.16.232.8.
```

Die Anwendung sendet:

```
GET /diagnosis4?PW=wut&
```

Das Web-IO antwortet:

```
diagnosis4;Watchdog Timer abgelaufen!
```

Die Anwendung sendet:

```
GET /diagnosis5?PW=wut&
```

Das Web-IO antwortet:

```
diagnosis5;TCP Client Alarm: Server nicht erreicht.
```

**Löschen des Fehlerspeichers**
Die Liste der zur Laufzeit aufgetretenen Fehler bleibt im Web-IO gespeichert, auch wenn der Fehler ggf. zum Zeitpunkt der Abfrage nicht mehr vorliegt.

Um den Fehlerspeicher zu löschen wird das Kommando

```
GET /diagclear?PW=password&
```

verwendet.

Das Web-IO antwortet mit

```
diagnosis;0000;00000000;00000000
```

wenn keine aktuellen Fehler mehr anliegen.

*Bitte beachten Sie, dass die* GET HEADER ENABLE *Einstellung unter* Config >> Device Basic Settings >> HTTP *auch den Error und Diag-Meldungen einen Header voranstellt. Auch das Anhängen des 0-Characters wird bei den Fehlermeldungen beibehalten.*

**BINARY - Socketprogramme mit Binärstrukturen**
Das Web-IO Digital stellt zwei eigenständige Socketzugänge, *Binary 1* und *Binary 2*, für den binären Datenaustausch zur Verfügung. Beide können unabhängig voneinander genutzt und konfiguriert werden.

Ob das Web-IO auf dem jeweiligen BINARY Socket als TCP-Server, TCP-Client oder UDP-Peer arbeiten soll, hängt von der gewünschten Applikation ab.

Hier zunächst eine Übersicht, für welche Applikation das Web-IO in welchem Operationsmodus arbeitet.

- Eigene Socket-Anwendung (ohne Passwortschutz) und Anwendungen, die ursprünglich für den W&T Digital- EA-Com-Server 50xxx programmiert wurden.
  - TCP-Server
  - TCP-Client

- UDP-Peer

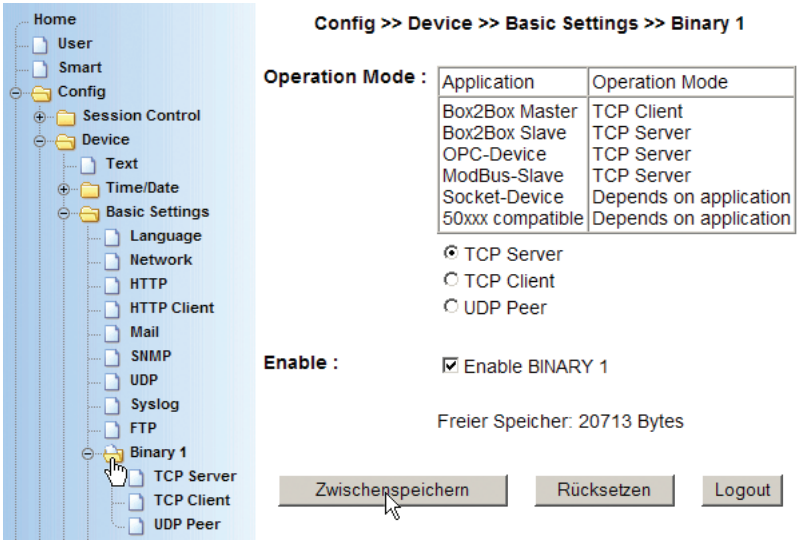
- Eigene Socket Anwendung (binär mit Passwortschutz)
  - TCP-Server
  - TCP-Client
  - UDP-Peer

**Binärer Socket-Zugang und 50xxx-Mode**
In diesem Kapitel wird gezeigt, wie das Web-IO Digital aus eigenen professionellen Anwendungen heraus über Sockets mit binären Strukturen angesprochen werden kann

**Festlegen des Operationsmodus**
Zunächst muss festgelegt werden, ob das Web-IO in Ihrer Anwendung TCP-Client, TCP-Server oder UDP-Peer sein soll.

Wählen Sie im Navigationsbaum *Config >> Device >> Basic Settings >> Binary 1* wenn Sie den Operationsmodus für den Zugang über *Binary 1* konfigurieren möchten.

Notwendige Zugriffsrechte: *Administrator*




Nach Wahl des gewünschten Modus und Setzen von *Enable Binary* übergeben Sie die Einstellung durch Klick auf den *Zwi-*

*schenspeichern*-Button an das Web-IO.

Für den Zugriff aus eigenen Anwendungsprogrammen stehen dem Entwickler zwei Stufen der Socketprogrammierung zur Verfügung.

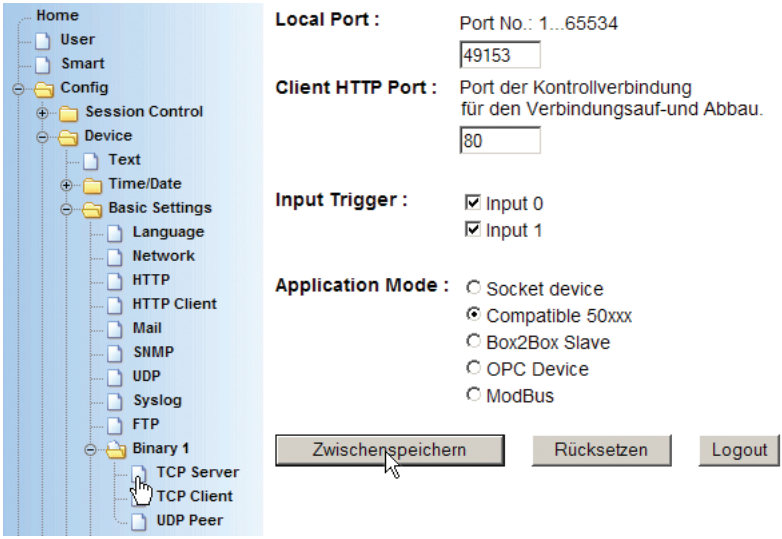
1. Passwortgeschützter Zugang
2. 50xxx-Mode kompatibel zu den älteren W&T Digital EA-Com-Servern

Beide Zugänge arbeiten mit den gleichen Binär-Strukturen und unterscheiden sich sonst nur durch den fehlenden Passwortschutz im 50xx-Modus

**Das Web-IO als Socket-Server**
Um das Web-IO als Socket-Server zu betreiben, müssen noch einige weitere Einstellungen vorgenommen werden.

Wählen Sie im Navigationsbaum *Config >> Device >> Basic Settings >> Binary 1>> TCP Server*

Notwendige Zugriffsrechte: *Administrator*



**Local Port**
Der Lokale Port des Web-IO ist werksseitig auf 49153 voreingestellt. Wenn Ihre Anwendung einen anderen Lokalen Port für das Web-IO erfordert, tragen Sie im Feld *Local Port* die gewünschte Port Nummer ein.

**Client HTTP Port**
Hat nur in den Betriebsarten OPC und Box2Box Bedeutung und legt den HTTP Port fest, auf den die Steuerverbindung zum OPC-Server bzw. zur Slave-Box aufgebaut werden soll.

Wenn nicht anders vorgegeben, sollte hier immer Port 80 verwendet werden.

**Input Trigger**
Markieren Sie hier die Inputs, deren Zustandsänderung eine Datensendung an die Client-Anwendung auslösen soll (wichtig für ereignisgesteuerte Anwendungen).

**Application Mode**
Wählen Sie hier:

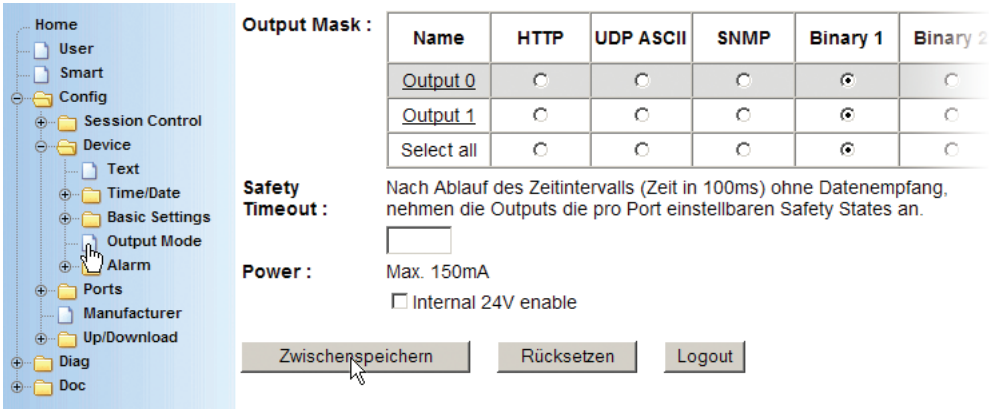
- *Compatible 50xxx* - für Anwendungen, die ohne Passwortschutz arbeiten sollen oder für Anwendungen die für die älteren Digital EA-Com-Server programmiert wurden.
- *Socket device* - wenn der Zugriff auf das Web-IO passwortgeschützt erfolgen soll.

Nachdem alle Einstellungen vorgenommen wurden, übergeben Sie diese durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Des Weiteren, müssen die benutzten Outputs noch für den Binary-Modus freigeschaltet werden.

Wählen Sie nun im Navigationsbaum *Config >> Device >> Output Mode* und markieren den gewünschten Binary Zugang.

Notwendige Zugriffsrechte: *Administrator*




Nachdem alle Eingaben vorgenommen wurden, übergeben Sie die Einstellung durch Klick auf den *Logout*-Button an das Web-IO. Durch Klick auf den *Speichern-Button* werden die Einstellungen aktiviert.

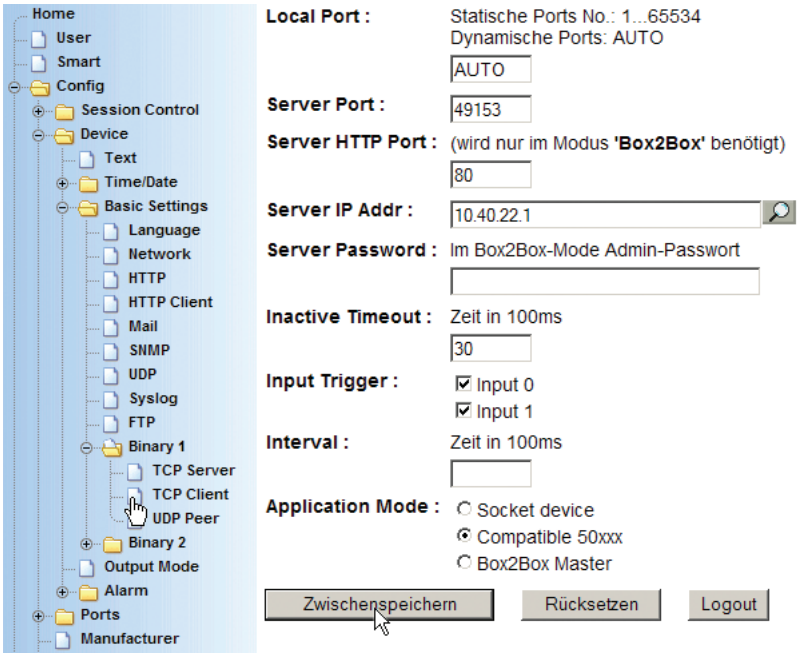
*Alle für* Binary 1 *gezeigten Konfigurationsmöglichkeiten können auch für* Binary 2 *genutzt werden.*

**Das Web-IO als Socket-Client**

Um das Web-IO als Socket-Client zu betreiben, müssen noch einige weitere Einstellungen vorgenommen werden.

Wählen Sie im Navigationsbaum *Config >> Device >> Basic Settings >> Binary 1>> TCP Client*

Notwendige Zugriffsrechte: *Administrator*




**Local Port**

Der lokale Port des Web-IO ist werksseitig auf 49153 voreingestellt. Wenn Ihre Anwendung einen anderen Lokalen Port für das Web-IO erfordert, tragen Sie im Feld *Local Port* die gewünschte Port Nummer ein.

**Server Port**

Tragen Sie hier die Portnummer ein, auf der die Server-Anwendung die Verbindung entgegennimmt.

**Server HTTP Port**

Hat nur in den Betriebsarten Box2Box Bedeutung und legt den HTTP Port fest, auf den die Steuerverbindung zur Slave-Box aufgebaut werden soll.

Wenn nicht anders vorgegeben, sollte hier immer Port 80 verwendet werden.

**Server IP Addr**

Tragen Sie hier die IP-Adresse des Servers ein.

**Server Password**
Ein Server-Passwort muss nur dann eingetragen werden, wenn das Web-IO als Box-to-Box Master arbeiten soll oder als TCP-Client auf ein weiteres Web-IO im Server Mode zugreifen soll. Dazu mehr im Kapitel Box-to-Box

**Inactive Timeout**
Hier wird der Timer konfiguriert, nach dessen Ablauf das Web-IO die TCP-Verbindung schließt. Der Wert ist dezimal und in 100ms Schritten anzugeben. Der Timer wird bei aktiver Netzwerkverbindung zurückgesetzt, wenn Daten ausgetauscht werden.

Beispiel: Der Wert 10 entspricht einer Sekunde. Wird für eine Sekunde kein Datentransfer erkannt, beendet das Web-IO die Verbindung

Wird kein Wert eingetragen, ist der automatische Verbindungsabbau deaktiviert.

**Input Trigger**
Markieren Sie hier die Inputs, deren Zustandsänderung als Trigger für den TCP-Verbindungsaufbau bzw. Datensendung an den Server dienen sollen (wichtig für ereignisgesteuerte Anwendungen).

**Interval**
Wenn der Zustand der Inputs zyklisch an die Server-Applikation übermittelt werden soll, kann hier das Intervall in 100ms Schritten eingetragen werden.

Beispiel: Der Wert 300 entspricht 30 Sekunden.

*Bitte beachten Sie, dass bei Verbindungen über kostenpflichtige Wahlverbindungen ein zu kleines Intervall dazu führen kann, dass diese Verbindung nicht getrennt wird und dadurch permanente Gebühren entstehen!*

**Mode**
Wählen Sie hier:

- *Socket Device*- Wenn der Zugriff auf ein weiteres Web-IO im Server-Modus passwortgeschützt erfolgen soll.
- *Compatible 50xxx* - Wenn Sie mit Server-Anwendungen kommunizieren möchten, die für die älteren Digital EA-Com-Server programmiert wurden. Diesen Modus können Sie auch für neue Anwendungen nutzen, die ohne Passwortschutz arbeiten sollen.

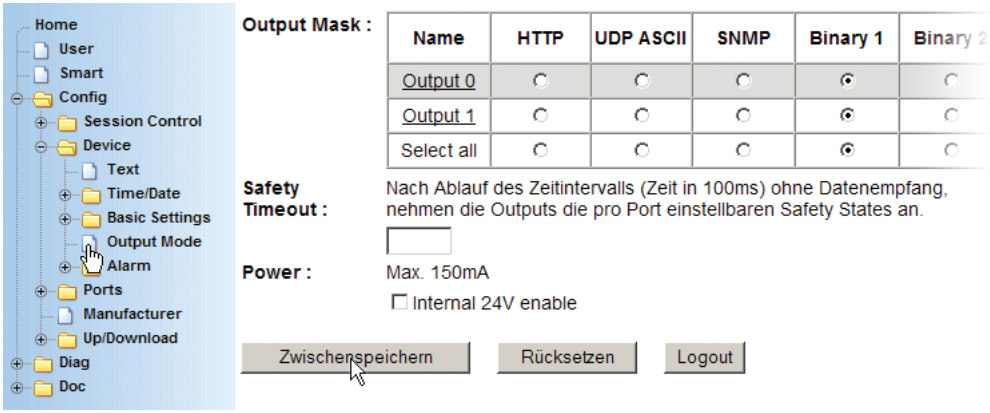
Auf den Modus *Box2Box Master* wird im Kapitel *Box-to-Box* näher eingegangen.

Nachdem alle Einstellungen vorgenommen wurden, übergeben Sie diese durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Des Weiteren müssen die benutzten Outputs noch für den Binary-Modus freigeschaltet werden.

Wählen Sie nun im Navigationsbaum *Config >> Device >> Output Mode* und markieren den gewünschten Binary-Zugang.

Notwendige Zugriffsrechte: *Administrator*



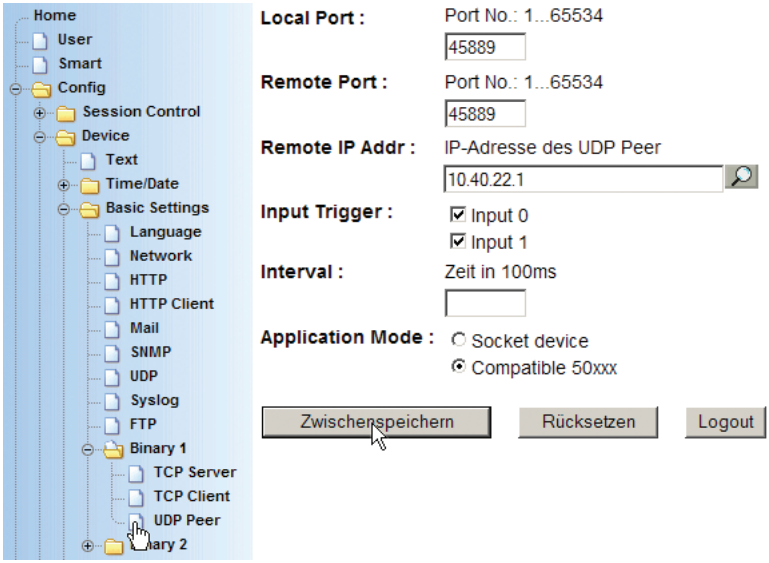

Nachdem alle Eingaben vorgenommen wurden, übergeben Sie die Einstellung durch Klick auf den *Logout*-Button an das Web-IO. Durch Klick auf den *Speichern-Button* werden die Einstellungen aktiviert.

### Das Web-IO als UDP-Peer

Um das Web-IO als UDP-Peer zu betreiben, müssen noch einige weitere Einstellungen vorgenommen werden.

Wählen Sie im Navigationsbaum *Config >> Device >> Basic Settings >> Binary 1>> UDP Peer*

Notwendige Zugriffsrechte: *Administrator*




### Local Port

Der Lokale Port des Web-IO ist werksseitig auf 45889 voreingestellt. Wenn Ihre Anwendung einen anderen lokalen Port für das Web-IO erfordert, tragen Sie im Feld *Local Port* die gewünschte Port Nummer ein.

### Remote Port

Tragen Sie hier die Portnummer ein, auf der die UDP-Anwendung, mit der das Web-IO kommunizieren soll, die Daten entgegennimmt.

### Remote IP Addr

Tragen Sie hier die IP-Adresse des Kommunikationspartners ein.

**Input Trigger**
Markieren Sie hier die Inputs, deren Zustandsänderung als Trigger für das Versenden eines UDP-Datagrammes dienen sollen (wichtig für ereignisgesteuerte Anwendungen).

**Interval**
Wenn der Zustand der Inputs zyklisch an den Kommunikationspartner übermittelt werden soll, kann hier das Interval in 100ms Schritten eingetragen werden.

Beispiel: Der Wert 300 entspricht 30 Sekunden.

*Bitte beachten Sie, dass bei Verbindungen über kostenpflichtige Wahlverbindungen ein zu kleines Intervall dazu führen kann, dass diese Verbindung nicht getrennt wird und dadurch permanente Gebühren entstehen!*

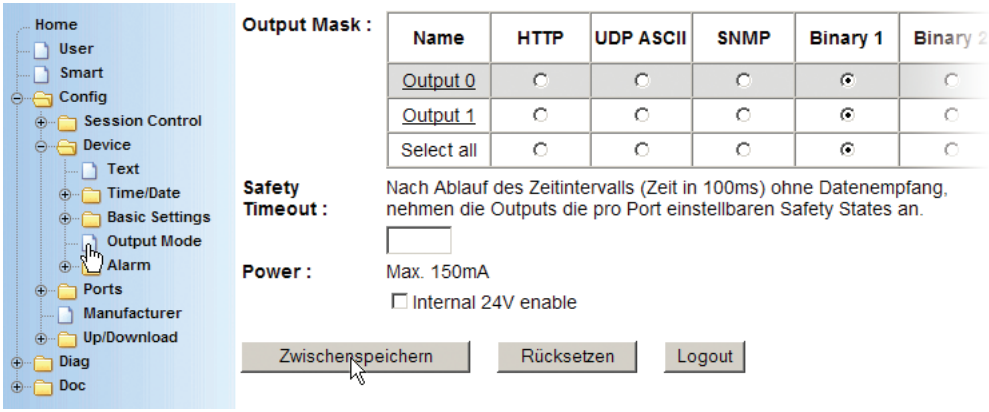
**Application Mode**
In der Konfiguration als UDP-Peer besteht kein Unterschied zwischen den Modi Default und Compatible 50xxx.

Nachdem alle Einstellungen vorgenommen wurden, übergeben Sie diese durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Des Weiteren müssen die benutzten Outputs noch für den Binary-Modus freigeschaltet werden.

Wählen Sie nun im Navigationsbaum *Config >> Device >> Output Mode* und markieren den gewünschten Binary Zugang.

Notwendige Zugriffsrechte: *Administrator*




Nachdem alle Eingaben vorgenommen wurden, übergeben Sie die Einstellung durch Klick auf den *Logout*-Button an das Web-IO. Durch Klick auf den *Speichern-Button* werden die Einstellungen aktiviert.

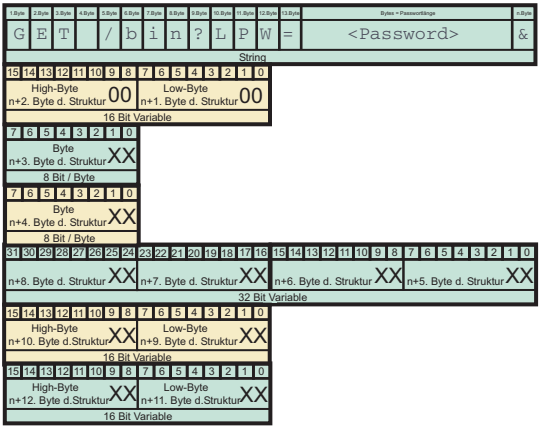
**Passwort-Schutz**

Wie bereits mehrfach angesprochen, bietet das Web-IO im TCP-Server Modus die Möglichkeit, den Zugang durch die Anwendung mit einem Passwort zu schützen.

Bevor die eigentliche Verbindung zum Web-IO aufgebaut wird, muss die hier definierte Struktur *BinInfo* über eine separate TCP-Verbindung an den HTTP Port (ab Werk Port 80 voreingestellt) des Web-IO gesendet werden.

Für die Antwort benutzt das Web-IO ebenfalls die Struktur *BinInfo*.

**Die BinInfo Struktur**




| | |
|---|---|
| **HTTPLogin** | String |
| **Dummy** | Word (16 Bit)<br>Immer = 0 |
| **Type** | Byte (8 Bit)<br>Type der Anfrage |
| **SubType** | Byte (8 Bit)<br>Subtype der Anfrage |
| **IPAddress** | Longword (32 Bit)<br>IP-Adresse des Absenders |
| **SrcPort** | Word (16 Bit)<br>TCP-Port des Absenders |
| **DstPort** | Word (16 Bit)<br>TCP-Port des Empfänders |

Die einzelnen Variablen der Struktur werden ausgefüllt wie folgt:

**HTTPLogin[n]**
Ist ein Bytefeld bzw. String, der sich aus einem Loginstring und dem verwendeten Administrator-Passwort zusammensetzt.

```
GET /bin?LPW=<Administator Passwort>&
```

n steht für die Anzahl der verwendeten Bytes und entspricht 14 + Länge des Passwortes. Die Länge des Passwortes ist auf maximal 31 Zeichen beschränkt.

In der Antwort des Web-IO ist HTTPLogin immer 8 Zeichen lang und enthält folgenden String:

```
GET /bin
```

**Dummy**
Trennzeichen zwischen dem ASCII- und dem Binär-Teil der Struktur.

Ist immer = 0x00

**Type**
Bestimmt die Art, in der der Binary-Modus genutzt werden soll.

Die Anwendung muss hier 0x04 eintragen, um eine TCP-Verbindung einzuleiten.

Das Web-IO trägt in seiner Antwort

0x02 ein, wenn die Verbindungsanforderung akzeptiert wurde.

0x03 ein, wenn die Verbindungsanforderung abgelehnt wurde.

**SubType**
Gibt nähere Einzelheiten zum Status der Verbindungsanforderung.

Die Anwendung sendet immer 0x00.

Das Web-IO antwortet mit

```
0x01 BINSUBTYPE_OK,     // wenn die Verbindungsanforderung akzeptiert wurde.
0x02 BINSUBTYPE_NO_ACCESS,// wenn bereits eine Verbindung besteht
0x04 BINSUBTYPE_WAIT,   // wenn die Verbindung erst nach einem
                  // Timeout hergestellt werden darf
0x07 BINSUBTYPE_PW_MISMATCH,        // bei falschem Passwort
0x08 BINSUBTYPE_DEST_PORT_MISMATCH,   // bei falschem Destination Port
0x09 BINSUBTYPE_MODUS_MISMATCH,     // bei falschem Modus
```

Wenn 0x01 oder 0x04 empfangen wurde, kann die eigentliche Datenverbindung geöffnet werden.

**SrcPort**
Die Client-Anwendung trägt hier immer eine 0 ein.

Das Web-IO gibt hier den geöffneten Server-Port zurück (z.B. 49153 für Binary 1). Bei gescheitertem Login-Versuch trägt das Web-IO 80 ein.

**DstPort**
**Die Client-Anwendung trägt hier ein, zu welchem Port Verbindung aufgenommen werden soll (z.B. 49153 für Binary 1 oder 49154 für Binary 2).**

**Das Web-IO gibt immer 0 zurück**

Die Verbindung, über die die BinInfo-Struktur ausgetauscht wurde, wird automatisch vom Web-IO beendet.

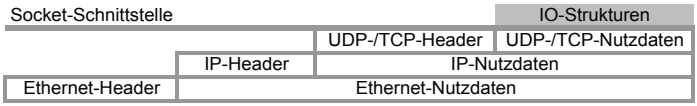
**BINARY - Die IO-Strukturen**
Um eine einfache Kommunikation zwischen dem Anwenderprogramm auf dem Rechner und dem Web-IO zu ermöglichen, gibt es eine übersichtliche Menge von Strukturen (Variablenfeldern), die das Format und den Inhalt der Daten definiert, die zwischen Anwenderprogramm und Web-IO ausgetauscht werden.

Für folgende Funktionen werden IO-Strukturen angeboten:

- Lesen der Inputs
- Lesen der Inputs und Outputs
- Setzen der Outputs
- Lesen der Counter
- Zurücksetzen der Counter
- Parametrieren der zyklischen und automatischen Benachrichtigung bei Zustandsänderung

Das Anwenderprogramm nutzt die einfach zu handhabende Socket-Schnittstelle (Windows: WinSock, UNIX, Linux: Berkley Sockets), um die Daten in Form dieser IO-Strukturen mit dem Web-IO über das Netzwerk per TCP/IP auszutauschen.

*Die IO-Strukturen sind unabhängig von dem gewählten Netzwerkprotokoll (TCP oder UDP).*




Für welches der beiden Protokolle, UDP oder TCP man sich entscheidet, ist von der Art der Applikation abhängig. Beide

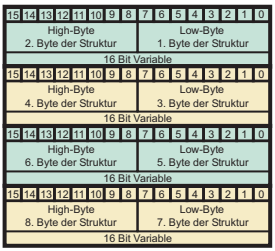
Protokolle bieten Vor- und Nachteile, die man in Abhängigkeit der zu erstellenden Anwendung gegeneinander abwägen muss

*Hilfe zur Socket-Programmierung inklusive Grundlagen TCP/IP finden Sie in kurzer und übersichtlicher Form in unserem Handbuch "Fit in 1 Tag für TCP/IP-Sockets". Programmbeispiele für Client-/Server-Applikationen unter TCP/IP finden Sie auf unserer Homepage im Internet unter http://www.wut.de.*

**Definition der IO-Strukturen**

Um den Inhalt eines Paketes eindeutig identifizieren und auswerten zu können, müssen im BINARY-Modus alle Daten in Form dieser IO-Strukturen an das Web-IO gesendet werden, unabhängig davon, ob die Betriebsart *50xxx-compatible* oder *Socket Device* benutzt wird.

Alle Strukturen beginnen mit dem gleichen Header, der aus den folgenden 4 WORDs (16bit_Integer) besteht:




| | |
|---|---|
| **Start_1** | Word (16 Bit)<br>Immer = 0 |
| **Start_2** | Word (16 Bit)<br>Immer = 0 |
| **StructType** | Word (16 Bit)<br>Strukturtype |
| **StructLength** | Word (16 Bit)<br>Länge in Bytes |

**Start_1, Start_2**

*Start_1* und *Start_2* sind aus Gründen der Kompatibilität zu Vorgängermodellen vorhanden, werden aber nicht benutzt. Beide Werte sind immer 0.

**StruktTyp**

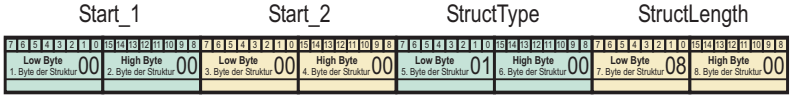
Der Wert *struct_typ* identifiziert, um welche Struktur es sich handelt. Sowohl Die PC-Anwendung als auch das Web-IO entscheiden bei Datenempfang anhand des Wertes *struct_typ,* wie die Struktur ausgewertet werden soll.

**StructLength**
*length* gibt die Gesamtlänge der Struktur in Bytes an, also inklusive der ersten 4 WORDs.

Achtung: Für alle IO-Strukturen gilt:

Ein WORD entspricht 16bit_integer. (ohne Vorzeichen)
Ein BYTE entspricht einem Byte (8Bit)
Ein LONG entspricht einem 32bit_integer (ohne Vorzeichen)

Beim Senden und Empfangen gilt für alle Struktur-Variablen: Low-Byte first.



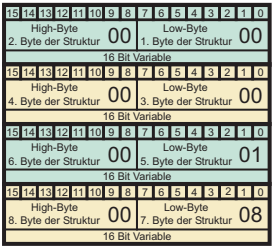

*In den Beispielen sind alle Zahlen hexadezimal dargestellt!*

### 5.6.3 Arbeiten mit den IO-Strukturen

Im nächsten Kapitel werden die einzelnen Strukturen erläutert und die entsprechenden Werte der Variablen *Start_1, Start_2, struct_typ* und *length*, mit denen jedes Paket beginnt, angegeben.

**Die IO-Struktur Read Register**
Das Senden dieser Struktur an das Web-IO veranlasst dieses, den Status der Inputs an das Anwenderprogramm zu senden. Das Paket besteht nur aus diesen vier WORDs. Diese Struktur wird nur vom Anwenderprogramm verwendet und das Web-IO reagiert immer mit dem Senden der Struktur WriteRegister.

**Die ReadRegister Struktur**



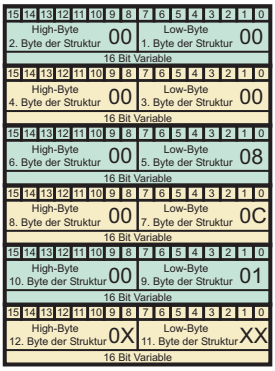

| | |
|---|---|
| **Start_1** | Word (16 Bit)<br>Immer = 0 |
| **Start_2** | Word (16 Bit)<br>Immer = 0 |
| **StructType** | Word (16 Bit)<br>Strukturtype |
| **StructLength** | Word (16 Bit)<br>Länge in Bytes |

## Die IO-Struktur WriteRegister

Diese Struktur dient zum Übertragen des Zustandes der Inputs oder Outputs beim Web-IO 12x Digital Input, 12x Digital Output. Sendet das Anwenderprogramm diese Struktur an das Web-IO, setzt das Web-IO die Outputs entsprechend des in *value* übergebenen Wertes.

Sendet das Web-IO diese Struktur an das Anwenderprogramm, hat *value* den dem Input-Status entsprechenden Wert.

**Die WriteRegister Struktur**




| | |
|---|---|
| **Start_1** | Word (16 Bit)<br>Immer = 0 |
| **Start_2** | Word (16 Bit)<br>Immer = 0 |
| **StructType** | Word (16 Bit)<br>Strukturtype |
| **StructLength** | Word (16 Bit)<br>Länge in Bytes |
| **Amount** | Word (16 Bit)<br>Immer = 1 |
| **Value** | Word (16 Bit)<br>Binäre Entsprechung des Inputstatus |

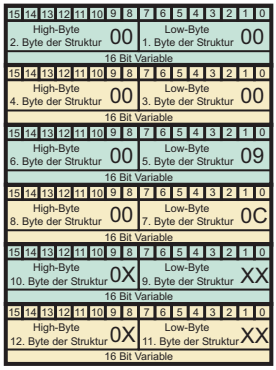
*Bei Web-IO mit Relaisausgängen, muss zwischen zwei Schaltvorgängen an einem Output ein Pause von min. 200ms eingehalten werden!*

## Die IO-Struktur SetBit

Diese Struktur ermöglicht das Setzen einzelner Outputs beim Web-IO. Wird .B. nicht der gesamte Prozessstatus im Anwenderprogramm abgebildet, können einzelne Outputs gesetzt

werden, ohne den Wert der anderen zu verändern. Die Bits 0..11 in *set_bits* und *value* korrespondieren mit den entsprechenden Outputs. Diese Struktur wird nur vom Anwenderprogramm verwendet.

**Die SetBit Struktur**




| | |
|---|---|
| **Start_1** | Word (16 Bit)<br>Immer = 0 |
| **Start_2** | Word (16 Bit)<br>Immer = 0 |
| **StructType** | Word (16 Bit)<br>Strukturtype |
| **StructLength** | Word (16 Bit)<br>Länge in Bytes |
| **Mask** | Word (16 Bit)<br>Die Bits, die verändert werden sollen = 1 |
| **Value** | Word (16 Bit)<br>Binäre Entsprechung des Inputstatus |

Beispiel.:

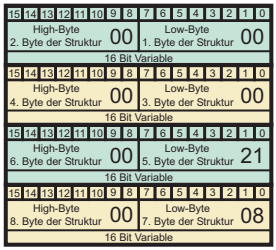
```
set_bits = 0x0124 / value = 0x0104
```

Output 2 und Output 8 (Zählweise Output0..1) werden auf ON gesetzt und Output 5 auf OFF. Alle anderen Outputs werden nicht verändert.

### Die IO-Struktur RegisterRequest

Diese Struktur sendet das Anwenderprogramm an das Web-IO, um den Inhalt von Inputs und Outputs im Überblick lesen zu können. Das Web-IO antwortet immer mit der E/A-Struktur *RegisterState*

**Die RegisterRequest Struktur**

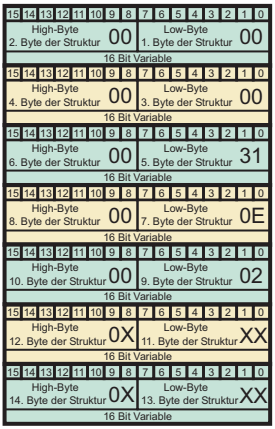



| | |
|---|---|
| **Start_1** | Word (16 Bit)<br>Immer = 0 |
| **Start_2** | Word (16 Bit)<br>Immer = 0 |
| **StructType** | Word (16 Bit)<br>Strukturtype |
| **StructLength** | Word (16 Bit)<br>Länge in Bytes |

### Die IO-Struktur RegisterState

Das Web-IO übermittelt mit dieser Struktur den Inhalt der Inputs und Outputs. Diese Struktur wird nur gesendet, wenn das Anwenderprogramm die Struktur *Register Request* an das Web-IO gesendet hat.

**Die RegisterState Struktur**



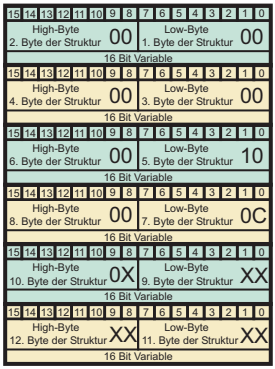

| | |
|---|---|
| **Start_1** | Word (16 Bit)<br>Immer = 0 |
| **Start_2** | Word (16 Bit)<br>Immer = 0 |
| **StructType** | Word (16 Bit)<br>Strukturtype |
| **StructLength** | Word (16 Bit)<br>Länge in Bytes |
| **DriverID** | Word (16 Bit)<br>Immer = 2 |
| **InputValue** | Word (16 Bit)<br>Binäre Entsprechung des Inputstatus |
| **OutputValue** | Word (16 Bit)<br>Binäre Entsprechung des Outputstatus |

### Die IO-Struktur SendMode

Mit dieser Struktur werden die Triggerbedingungen festgelegt, mit denen das Web-IO den Status der Inputs an das Anwenderprogramm sendet. Prinzipiell gibt es drei Möglichkeiten, die jedoch alle miteinander kombinierbar sind

1. Das Anwenderprogramm pollt das Web-IO durch Senden der READ-Struktur.
2. Das Web-IO sendet die WriteRegister - Struktur mit dem Status der Inputs in einem konfigurierbarem Zeitintervall.
3. Das Web-IO sendet die WriteRegister - Struktur mit dem Status der Inputs nach Zustandsänderung der konfigurierten Inputs.

**Die SendMode Struktur**

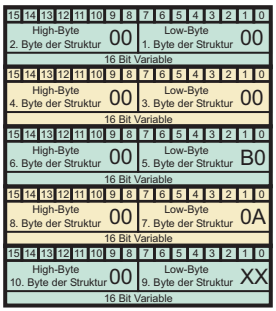



| | |
|---|---|
| **Start_1** | Word (16 Bit)<br>Immer = 0 |
| **Start_2** | Word (16 Bit)<br>Immer = 0 |
| **StructType** | Word (16 Bit)<br>Strukturtype |
| **StructLength** | Word (16 Bit)<br>Länge in Bytes |
| **Mask** | Word (16 Bit)<br>Bits für Input Trigger = 1 |
| **Interval** | Word (16 Bit)<br>Intervall in 100ms für zyklische Input Status Übermittlung |

## Die IO-Struktur ReadCounter

Diese Struktur sendet das Anwenderprogramm an das Web-IO, um den Zählerstand von einem bestimmten Input-Counter anzufordern. Um welchen Input es geht, wird in der Variablen *counter_index* übergeben. Das Web-IO antwortet immer mit der Struktur *Counter*.

**Die ReadCounter Struktur**

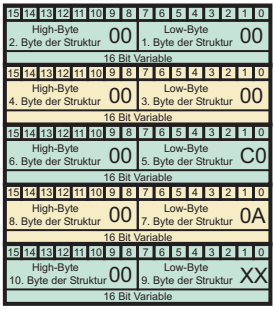



| | |
|---|---|
| **Start_1** | Word<br>Immer = 0 |
| **Start_2** | Word<br>Immer = 0 |
| **StructType** | Word<br>Strukturtype |
| **StructLength** | Word<br>Länge in Bytes |
| **CounterIndex** | Word<br>Nummer des Inputs, dessen Counter gelesen werden soll |

## Die IO-Struktur ReadClearCounter

Diese Struktur sendet das Anwenderprogramm an das Web-IO, um den Zählerstand von einem bestimmten Input-Counter anzufordern und den Zähler danach sofort auf 0 zu setzen. Um welchen Input es geht, wird in der Variablen *counter_index* übergeben. Das Web-IO antwortet immer mit der Struktur *Counter*.

**Die ReadClearCounter Struktur**

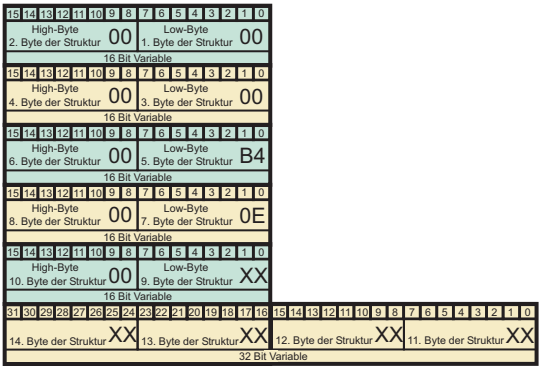



| | |
|---|---|
| **Start_1** | Word<br>Immer = 0 |
| **Start_2** | Word<br>Immer = 0 |
| **StructType** | Word<br>Strukturtype |
| **StructLength** | Word<br>Länge in Bytes |
| **CounterIndex** | Word<br>Nummer des Inputs, dessen Counter gelöscht werden soll |

## Die IO-Struktur Counter

Das Web-IO übermittelt mit dieser Struktur den Zählerstand des in *counter_index* angegebenen Input-Counters.

**Die Counter Struktur**

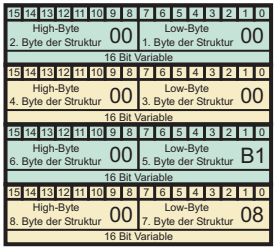



| | |
|---|---|
| **Start_1** | Word<br>Immer = 0 |
| **Start_2** | Word<br>Immer = 0 |
| **StructType** | Word<br>Strukturtype |
| **StructLength** | Word<br>Länge in Bytes |
| **CounterIndex** | Word<br>Nummer des Counters |
| **CounterValue** | Longword<br>Inputzählerwert 1 |

## Die IO-Struktur ReadAllCounter

Diese Struktur sendet das Anwenderprogramm an das Web-IO, um die Zählerstände von allen Inputs in einem Datenpaket anzufordern. Das Web-IO antwortet immer mit der Struktur *AllCounter*.

**Die ReadAllCounter Struktur**

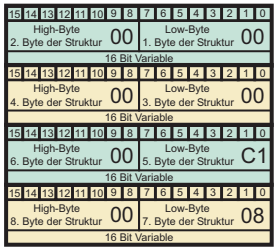



| | |
|---|---|
| **Start_1** | Word<br>Immer = 0 |
| **Start_2** | Word<br>Immer = 0 |
| **StructType** | Word<br>Strukturtype |
| **StructLength** | Word<br>Länge in Bytes |

## Die IO-Struktur ReadClearAllCounter

Diese Struktur sendet da Anwenderprogramm an das Web-IO, um die Zählerstände von allen Inputs in einem Datenpaket anzufordern und die Zähler danach sofort auf 0 zu setzen. Das Web-IO antwortet immer mit der Struktur *AllCounter*.

**Die ReadClearAllCounter Struktur**

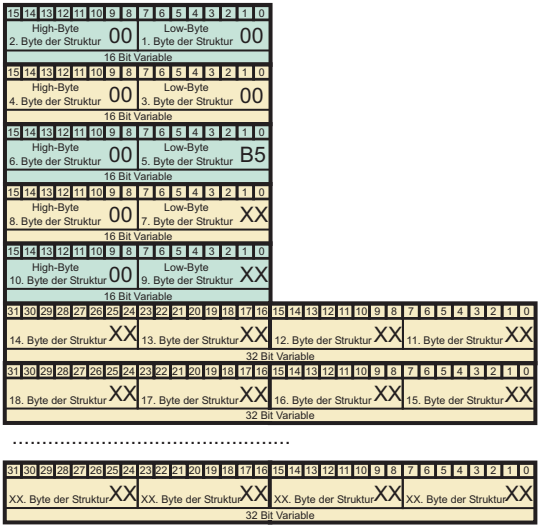



| | |
|---|---|
| **Start_1** | Word<br>Immer = 0 |
| **Start_2** | Word<br>Immer = 0 |
| **StructType** | Word<br>Strukturtype |
| **StructLength** | Word<br>Länge in Bytes |

## Die IO-Struktur AllCounte

Das Web-IO übermittelt mit dieser Struktur die Zählerstände aller Inputs auf einmal.

**Die AllCounter Struktur**

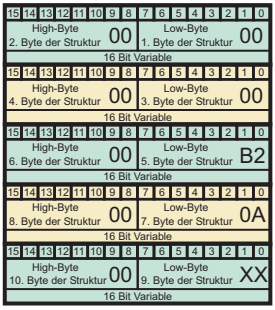



| | |
|---|---|
| **Start_1** | Word<br>Immer = 0 |
| **Start_2** | Word<br>Immer = 0 |
| **StructType** | Word<br>Strukturtype |
| **StructLength** | Word<br>Länge in Bytes |
| **CounterNoOf** | Word<br>Anzahl der Counter |
| **CounterValue1** | Longword<br>Inputzählerwert 1 |
| **CounterValue2** | Longword<br>Inputzählerwert 2 |
| **CounterValuen** | Longword<br>Inputzählerwert n |

## Die IO-Struktur ClearCounter

Diese Struktur sendet das Anwenderprogramm an das Web-IO, um den Zählerstand eines bestimmten Input-Counters auf 0 zurückzusetzen. Um welchen Input es geht, wird in der Variablen *counter_index* übergeben.

**Die ClearCounter Struktur**



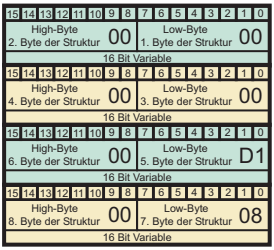

| | |
|---|---|
| **Start_1** | Word<br>Immer = 0 |
| **Start_2** | Word<br>Immer = 0 |
| **StructType** | Word<br>Strukturtype |
| **StructLength** | Word<br>Länge in Bytes |
| **CounterIndex** | Word<br>Nummer des Inputs, dessen Counter gelesen werden soll |

## Die IO-Struktur ReadDiagnosis

Stellt das Web-IO einen Kommunikations- oder Systemfehler fest, wird dieser auf der HTML-Seite *diag* aufgelistet und kann über den Browser ausgelesen werden. Da sich das Fehlermanagement via Browser für programmgesteuerte Applika-

tionen nicht immer anbietet, kann der Fehlerstatus des Web-IO über die Struktur *ReadDiagnosis* abgefragt werden

**Die ReadDiagnosis Struktur**



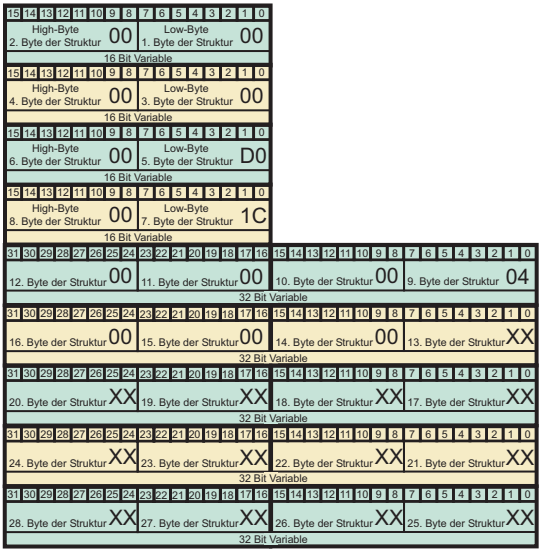

| | |
|---|---|
| **Start_1** | Word (16 Bit)<br>Immer = 0 |
| **Start_2** | Word (16 Bit)<br>Immer = 0 |
| **StructType** | Word (16 Bit)<br>Strukturtype |
| **StructLength** | Word (16 Bit)<br>Länge in Bytes |

Als Antwort sendet das Web-IO eine Struktur vom Typ Diagnosis

## Die IO-Struktur Diagnosis

Das Web-IO sendet die Struktur Diagnosis als Antwort auf die *ReadDiagnosis* Struktur.

**Die Diagnosis Struktur**




| | |
|---|---|
| **Start_1** | Word (16 Bit)<br>Immer = 0 |
| **Start_2** | Word (16 Bit)<br>Immer = 0 |
| **StructType** | Word (16 Bit)<br>Strukturtype |
| **StructLength** | Word (16 Bit)<br>Länge in Bytes |
| **Version** | Longword (32 Bit)<br>z.Zt. V4 |
| **ErrorCount** | Longword (32 Bit)<br>aktuelle Fehlerzahl |
| **ErrorBits1** | Longword (32 Bit)<br>binäre Fehler-Kodierung |
| **ErrorBits2** | Longword (32 Bit)<br>binäre Fehler-Kodierung |
| **ErrorBits3** | Longword (32 Bit)<br>binäre Fehler-Kodierung |

In der Variablen *ErrorCount* wird zurückgegeben, wieviele verschiedene Fehler zur Zeit im Fehlerspeicher sind. Insgesamt unterscheidet das Web-IO bis zu 92 verschiedene Fehlerzustände, wobei jedes gesetzte Bit der beiden Variablen

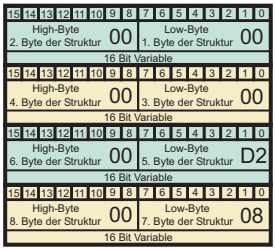
*ErrorBits1* bis *ErrorBits3* für eine Fehlerart steht.

Die genaue textliche Beschreibung kann über TCP Port 80 abgerufen werden. Eine nähere Beschreibung hierzu finden Sie im Kapitel Socketprogrammierung mit Kommandostrings.

### Die IO-Struktur ClearDiagnosis

Mit dieser Struktur wird der Fehlerspeicher des Web-IO gelöscht.

**Die ClearDiagnosis Struktur**

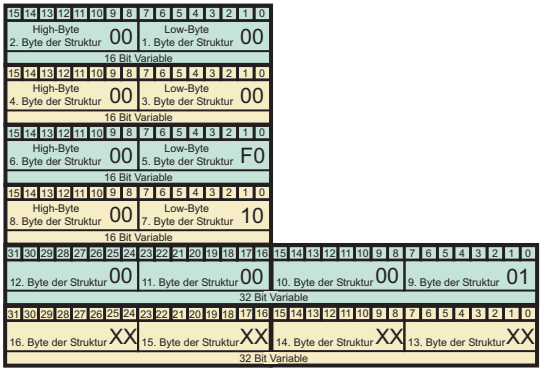



| | |
|---|---|
| **Start_1** | Word (16 Bit)<br>Immer = 0 |
| **Start_2** | Word (16 Bit)<br>Immer = 0 |
| **StructType** | Word (16 Bit)<br>Strukturtype |
| **StructLength** | Word (16 Bit)<br>Länge in Bytes |

### Die IO-Struktur Options

Diese Struktur dient dazu, im Web-IO bestimmte Optionen zu setzen. Dazu stehen 32 Bit in der Variablen *options* zur Verfügung.

**Die Options Struktur**




| | |
|---|---|
| **Start_1** | Word (16 Bit)<br>Immer = 0 |
| **Start_2** | Word (16 Bit)<br>Immer = 0 |
| **StructType** | Word (16 Bit)<br>Strukturtype |
| **StructLength** | Word (16 Bit)<br>Länge in Bytes |
| **Version** | Longword (32 Bit)<br>z.Zt. V4 |
| **Options** | Longword (32 Bit)<br>binäre Options-Kodierung |

In der vorliegenden Version des Web-IO wird nur das Bit 0 der Variablen *options* verwendet.

Bit 0 = 1 //das Web-IO sendet die RegisterState Struktur zurück, wenn ein Output gesetzt wird.

Bit 0 = 0 //das Web-IO gibt bei Setzen eines Output keine Rückmeldung.

*Um in der Rückmeldung zuverlässig den Zustand nach Setzen des/der Output(s) zu bekommen, sollte zwischen zwei Output-verändernden Zugriffen min. eine Zeit 150ms liegen.*

## 5.7 Wayback - Schalten per FTP-Login

Über die Standardbetriebsarten hinaus beinhaltet das ERP-Gateway-Modell 57638 zusätzlich das patentierte Wayback-Verfahren für den Zugriff auf die digitalen Ausgänge. Dieses ermöglicht es, Schaltbefehle über den Login-Namen eines FTP-Clients zu übermitteln. Hierdurch werden auch kommunikativ eingeschränkte Systeme wie z.B SAP oder anderen ERP/PPS-Anwendungen in die Lage versetzt, direkt auf die Ein-/Ausgänge zuzugreifen - ohne spezielle Treiber, spezielle Softwarepakete oder besondere Zugriffsrechte des Anwenders auf das Dateisystem.

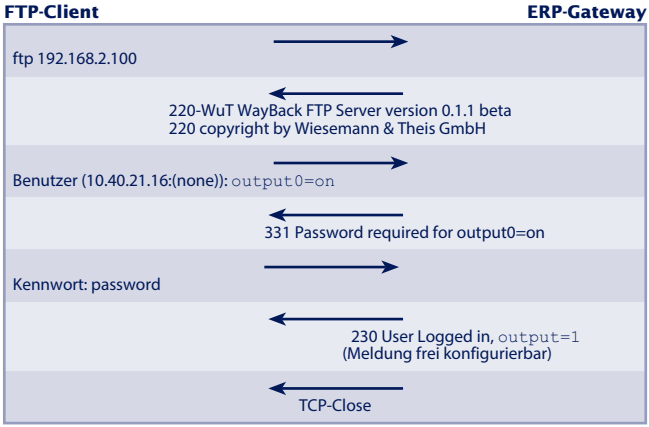
*Im Gegensatz zu den übrigen Web-IO Digital, ist beim Modell 57638 das Wayback FTP als Werkseinstellung vorkonfiguriert.*

### 5.7.1 Funktionsprinzip

Nach dem TCP-Verbindungsaufbau, sendet der Wayback-FTP-Server seine Begrüßungsmeldung an den FTP-Client. Als *User-Name* wird jetzt das Kommando zum Schalten der Ausgänge oder auch zum Lesen der Eingänge eingegeben. Mit der Authentifizierung des Kommandos durch das *Wayback-Passwort* wird der Befehl dann umgehend ausgeführt und zum Beispiel der angegebene Ausgang in den gewünschten Zustand geschaltet. Der Wayback-Server sendet die konfigurierbare Abschlussmeldung und schließt dann automatisch die Verbindung.

Zur Vermeidung gleichzeitiger, konkurrierender Zugriffe, ist der Wayback-Dienst exklusiv ausgelegt. Das heißt, nach dem Verbindungsaufbau durch einen Client, muss diese Verbindung zunächst durch eine gültige Kommando/Passwort-Kombination oder den Timeout beendet werden, bevor ein neuer Zugriff akzeptiert wird.

Ablaufbeispiel eines Schaltkommandos für Output 0:




### 5.7.2 Die Wayback-Kommandos

Die folgenden Kommandos können als FTP-User-Name an das ERP-Gateway übertragen werden. Groß-/Kleinschreibung wird ignoriert. Alle Antworten des ERP-Gateways werden mit der Variable *<MSG>* in der *Wayback Response* übermittelt.

**output[x]=[on|off]**

*x* Nummer des gewünschten Ausgangs, beginnend mit 0 für den ersten Ausgang eines ERP-Gateways
*on* Schaltet den Ausgang *x* ein
*off* Schaltet den Ausgang *x* aus

Das Kommando erlaubt das gezielte Ein-/Ausschalten einzelner Ausgänge des ERP-Gateways.

Beispiel:
*output0=on* schaltet den Ausgang 0 des ERP-Gateways in den Zustand ON

**output=[xxxx]**

*xxxx* Bitmuster des gewünschten Zustandes aller Ausgänge in hexadezimaler Schreibweise.

Beispiel:
*output=2* (binär = 0000 0010) schaltet den Ausgang 1 in

den Zustand ON, alle anderen verfügbaren Ausgänge gehen in den Status OFF
*output=3e* (binär = 0011 1110) schaltet die Ausgänge 1 bis 5 in den Zustand ON, alle anderen verfügbaren Ausgänge gehen in den Status OFF.

**output**
Das Kommando erlaubt das Auslesen des Zustandes der digitalen Ausgänge. Die Ausgabe erfolgt als hexadezimaler Wert des binären Zustandes aller Ausgänge mit der Variable *<MSG>* in der *Wayback Response.*

**input**
Das Kommando erlaubt das Auslesen des Zustandes der digitalen Eingänge. Die Ausgabe erfolgt als hexadezimaler Wert des binären Zustandes aller Eingänge mit der Variable *<MSG>* in der *Wayback Response.*

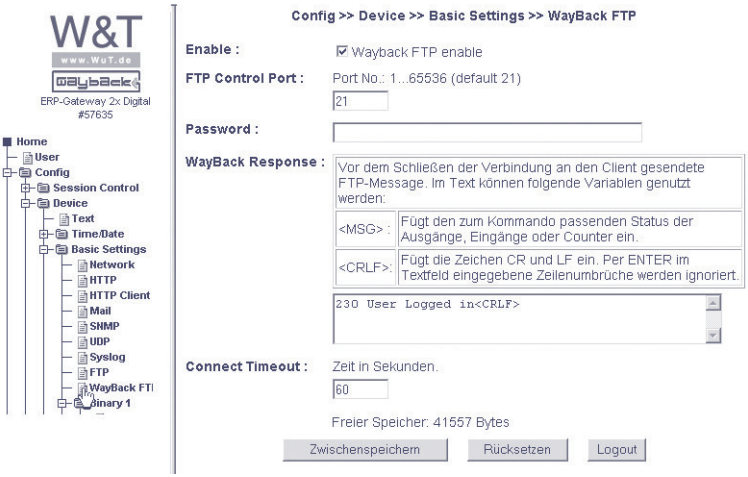
**counter[x]**
*x* Nummer des gewünschten Eingangs beginnend mit 0 für den ersten Ausgang des ERP-Gateways

Das Kommando erlaubt das Auslesen des Zählers des angegebenen digitalen Eingangs. Die Ausgabe erfolgt als dezimaler Wert mit der Variable *<MSG>* in der *Wayback Response.*

### 5.7.3 Freischalten des Wayback-Zugriffs

Notwendige Zugriffsrechte: *Administrator*

Wählen Sie im Navigationsbaum des ERP-Gateways *Config >> Device >> Basic Settings >> WayBack FTP*




Auf dieser Seite können die folgenden Einstellungen vorgenommen werden:

**Enable**

Aktivierung des Wayback-Dienstes. Ist diese Option deaktiviert, wird der Wayback-Server-Dienst nicht gestartet und Verbindungsversuche zum *FTP Control Port* werden abgewiesen.

**FTP Control Port**

Gibt den TCP-Server-Port an, unter welchem der Wayback-Dienst für FTP-Clients erreichbar ist. Der für FTP reservierte  und von den meisten FTP-Clients auch als Standard genutzte TCP-Port lautet 21.

**Password**

Das für die korrekte Ausführung des Kommandos am FTP-Client einzugebende Wayback-Passwort. Dieses ist *nicht* identisch mit dem *Admin*- oder *Operator*-Passwort des Gerätes.

**Wayback Response**

Abschließende Meldung, die nach dem Empfang des korrekten Wayback-Passwortes an den FTP-Client gesendet wird. Nach dieser Meldung schließt das ERP-Gateway au-

tomatisch die TCP-Verbindung. Innerhalb des Textes können folgende Variablen verwendet werden:

<MSG>
Das ERP-Gateway ersetzt diese Variable durch den zum vorherigen Kommando passenden Status der Ausgänge, Eingänge oder Counter. Die Ausgabe erfolgt als hexadezimaler Gegenwert des binären Zustandes aller Ein-/Ausgänge oder als Dezimalwert für die Counter. Wurde ein output-Kommando gesendet und die Ausgänge 0 und 1 sind beide eingeschaltet, lautet die Ausgabe zum Beispiel *output=3.* Sind an einem 57636 mit 6 Relais-Ausgängen die Ausgänge 0, 4 und 5 eingeschaltet, wird *output=31* zurückgegeben.

<CRLF>
Das ERP-Gateway ersetzt diese Variable durch die Zeichen CR (Carriage Return, 0x0d) und LF (Linefeed, 0x0a). Im Eingabefenster per Tastatur-ENTER eingefügte Zeilenumbrüche werden bei der Ausgabe *nicht* berücksichtigt.

**Connect Timeout**

Nach Ablauf der angegebenen Zeit in Sekunden ohne Aktivität des FTP-Clients, schließt das ERP-Gateway die bestehende TCP-Verbindung.

Nach Konfiguration der gewünschten Parameter werden die Änderungen durch Betätigung des Buttons *Zwischenspeichern* übernommen.

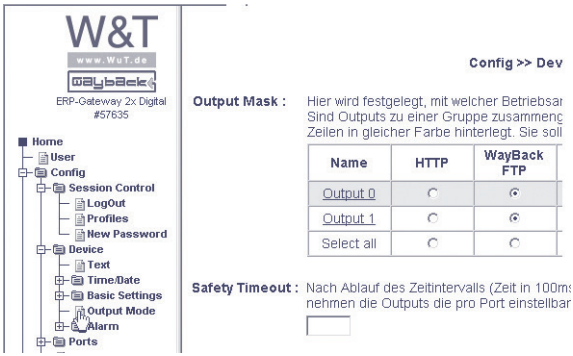
Wenn keine weiteren Konfigurationsänderungen mehr erfolgen sollen, müssen die neuen Einstellungen noch aktiviert werden. Benutzen Sie den *Logout*-Button oder wählen Sie im Menü *Config >> Session Control >> LogOut*

### 5.7.4 Aktivierung der Outputs für Waybac-Zugriff

Notwendige Zugriffsrechte: *Administrator*

*Die ERP-Gateways 57635 und 57636 sind ab Werk auf den Zugriff per Wayback-Verfahren vorkonfiguriert, so dass bei der Erstinbetriebnahme die hier beschriebene Aktivierung der Ausgänge entfallen kann.*

Wählen Sie im Navigationsbaum des ERP-Gateways *Config >> Device >> Output Mode.*




Aktivieren Sie hier die gewünschten Ausgänge, die per Wayback gesteuert werden sollen und betätigen dann den Button *Zwischenspeichern*.

Wenn keine weiteren Konfigurationsänderungen mehr erfolgen sollen, müssen die neuen Einstellungen noch aktiviert werden. Benutzen Sie den *Logout*-Button oder wählen Sie im Menü *Config >> Session Control >> LogOut*

# 6 Allgemeine Funktionen

- Time / Date
- Safety State - Zugriffsüberwachung
- Portkonfiguration
- Ändern der gerätespezifischen Anzeigen
- Upload / Download
- Diagnose und Test

## 6.1 Time / Date - Uhrzeit und Datum einstellen

Das Web-IO Digital verfügt über eine interne Uhr mit Datumsfunktionen.

### 6.1.1 Zeitzonen

Notwendige Zugriffsrechte: *Administrator*

Das Web-IO arbeitet intern mit der UTC-Zeit (Universal Time Coordinated), die auf der ganzen Welt als gemeinsame Zeitbasis genutzt wird. Die lokale Uhrzeit weicht, je nach dem in welcher Zeitzone man sich befindet, von der UTC ab. Die in Deutschland gültige Uhrzeit entspricht UTC + 1h (zur Winterzeit).

Wenn das Web-IO für den Einsatz in einer anderen Zeitzone bestimmt ist, kann der Offset entsprechend angepasst werden.

Wählen Sie hierzu im Navigationsbaum *Config >> Device >> Time/Date >> TimeZone:*

**UTCoffset**
Geben Sie hier die Differenz zwischen lokaler und UTC-Zeit im Format hh:mm ein. Im Eingabefeld kann zusätzlich ein Vorzeichen vorangestellt werden.

**Enable**
Aktivieren Sie *Apply Timezone,* damit der eingestellte Offset-Wert zum Tragen kommt.

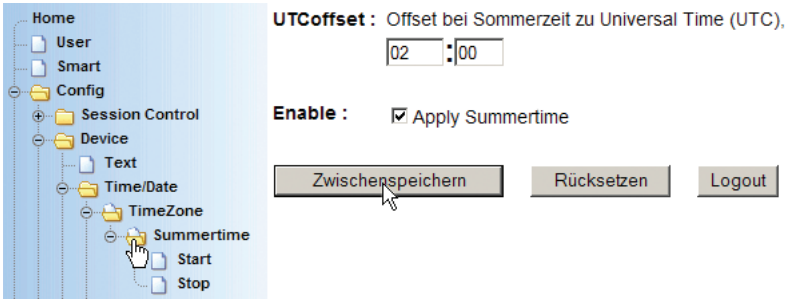
Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

### 6.1.2 Sommerzeit / Winterzeit

Notwendige Zugriffsrechte: *Administrator*

Das Web-IO unterscheidet automatisch zwischen Sommerzeit und Winterzeit.

Um die Eckwerte für die Sommerzeit/Winterzeit zu ändern wählen Sie im Navigationsbaum *Config >> Device >> Time/Date >> TimeZone >> Summertime:*




**UTCoffset**
Geben Sie hier die Differenz zwischen lokaler Sommerzeit und UTC Zeit im Format hh:mm ein. Im Eingabefeld kann zusätzlich ein Vorzeichen vorangestellt werden.

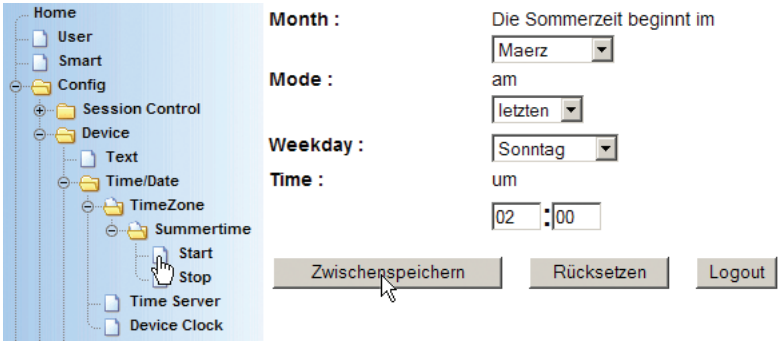
**Enable**
Markieren Sie *Apply Summertime*, damit der eingestellte Offset-Wert zum Tragen kommt.

Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

### 6.1.3 Beginn und Ende der Sommerzeit

Notwendige Zugriffsrechte: *Administrator*

Um einzustellen, wann die Sommerzeit beginnt, wählen Sie *Config >> Device >> Time/Date >> TimeZone >> Summertime >> Start bzw. Stop:*




**Month**
Wählen Sie hier den Monat aus, in dem auf Sommerzeit umgestellt wird.

**Mode**
Sollte die Sommerzeit nicht am letzten Sonntag des gewählten Monats beginnen bzw. enden, kann hier ein anderer Modus ausgewählt werden.

**Weekday**
Wenn die Umstellung nicht z.B. von Samstag auf Sonntag stattfindet, kann hier ein anderer Wochentag ausgewählt werden.

**Time**
Uhrzeit zu der die Umstellung stattfindet.

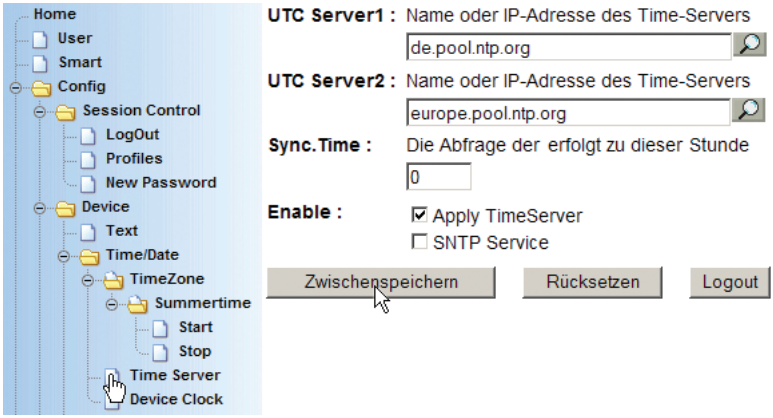
Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

### 6.1.4 Time-Server

Notwendige Zugriffsrechte: *Administrator*

Zur Synchronisation der internen Uhr des Web-IO, kann ein Time-Server angegeben werden. Das Web-IO ruft nach dem Einschalten die Uhrzeit beim Time-Server ab. Ferner findet einmal täglich ein Zeitabgleich statt.

Um die nötigen Einstellungen für den Time-Server einzugeben, wählen Sie im Navigationsbaum *Config >> Device >> Time/Date >> Time Server:*




**UTC-Server 1 + 2**
Bitte tragen Sie hier einen oder zwei Time-Server ein, die vom Standort des Web-IO erreichbar sind.

**Sync. Time**
Tragen Sie hier die volle Stunde ein, zu der der tägliche Abgleich mit dem Timeserver erfolgen soll.

Die Synchronisation mit dem Timeserver erfolgt zudem ca. 30 Sekunden nach Einschalten des Web-IO, bzw. nach einem Reset. Wird die Webseite des Web-IO sofort nach Neustart aufgerufen, wird noch die alte unsynchronisierte Uhrzeit angezeigt. Erst nach erneutem Aufruf der Seite wird die aktuelle Uhrzeit angezeigt.

**Enable**
Markieren Sie *Apply Timeserver,* damit die Uhrzeit künftig mit einem externen Timeserver synchronisiert wird.

Das Web-IO kann auch selber als SNTP-Timeserver fungieren (z.B. um andere Web-IO zu synchronisieren). In diesem Fall muss *SNTP Service* aktiviert werden.

Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

*Die werksseitig voreingestellten Adressen sind nur über einen Internetzugang erreichbar. Wenn für die Time-Server Namen (nicht IP-Adressen) angegeben wurden, muss in der Netzwerkkonfiguration ein DNS-Server angegeben sein.*

### 6.1.5 Manuelles Einstellen von Uhrzeit und Datum

Notwendige Zugriffsrechte: *Administrator* oder *Config*

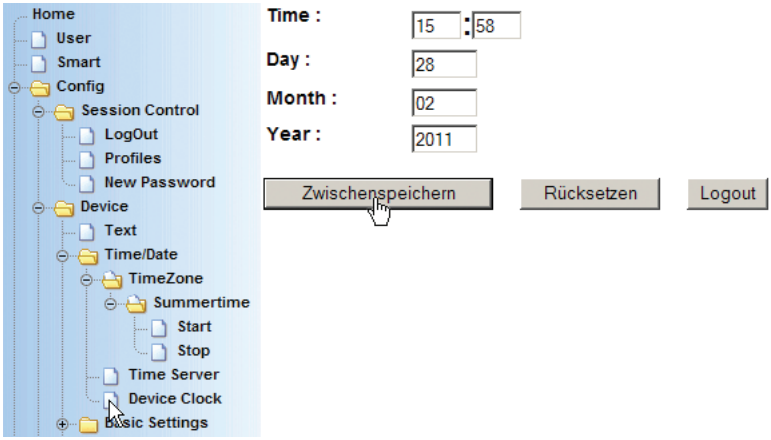
Wenn kein Zugang zu einem Time-Server vorhanden ist, kann die Uhrzeit unter *Config >> Device >> Time/Date >> Device Clock* auch manuell eingestellt werden.

**Time**
Uhrzeit im Format hh:mm. Eine sekundengenaue Einstellung ist nicht möglich.

**Day**
im Format dd

**Month**
im Format mm

**Year**
im Format yyyy




Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

### 6.1.6 Aktivieren aller Time-/Date-Einstellungen

Wenn alle den Alarm betreffenden Eingaben gemacht wurden, aktivieren Sie die neuen Einstellungen über *Config >> Session Control >> LogOut, Speichern.*

*Wenn Sie sich nach Aktivieren der Time-Server Synchronisation sofort wieder als Operator oder Administrator einloggen, kann es passieren, dass Ihnen nach einigen Sekunden der Zugriff verweigert wird.*

Das liegt daran, dass das Web-IO die Login-Rechte aus Sicherheitsgründen nach einer Stunde zurücksetzt. Hat das Web-IO zum Login-Zeitpunkt z.B. eine Zeiteinstellung von 12:00 Uhr 01.01.2002 (Werkseinstellung nach Power on) und der Time-Server setzt die Zeit auf 13:42 Uhr 26.06.2003 ist die Zeitspanne deutlich größer als eine Stunde und das Web-IO kündigt die Zugriffsrechte.

Nach erneutem Login kann dann die volle Stunde zur Konfiguration ausgeschöpft werden.

## 6.2 Safety Timeout / State - Zugriffsüberwachung

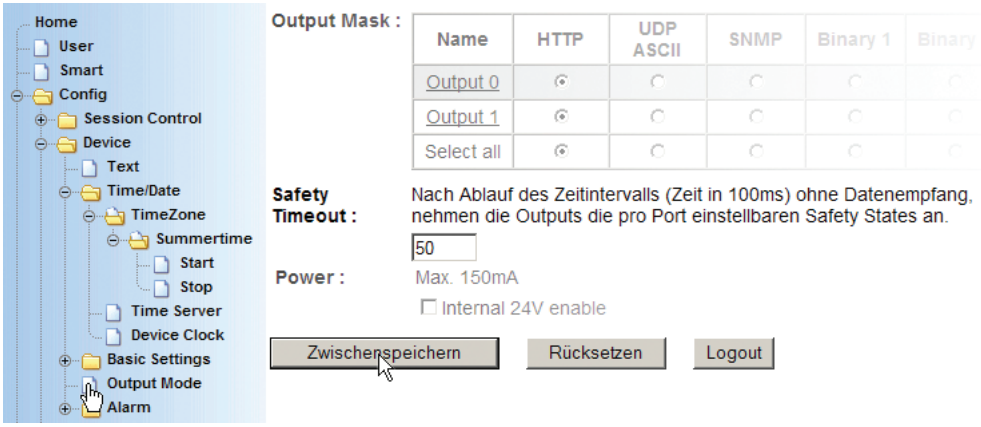
Komponenten die über ein Netzwerk miteinander kommunizieren, sind auf eine funktionierende Netzwerkinfrastruktur angewiesen. Auch wenn Netzwerke heute über ein Höchstmaß an Übertragungssicherheit verfügen, kann eine Störung nie ganz ausgeschlossen werden.

Um im Fall einer Netzwerkstörung ein kontrolliertes Verhalten beim Web-IO zu erreichen, gibt es den *Safety Timeout*. Der damit verbundene Watchdog überwacht, ob innerhalb eines konfigurierbaren Intervalls ein Netzwerkzugriff auf das Web-IO erfolgt ist. Der Watchdog Timer wird bei jedem Netzwerkzugriff zurückgesetzt und bringt die Outputs bei Erreichen der eingestellten Safety Timeout-Zeit in den konfigurierten Zustand.

### 6.2.1 Safety Timeout festlegen

Um den Safety Timeout zu konfigurieren wählen Sie im Navigationsbaum *Config >> Device >> Output Mode:*

Notwendige Zugriffsrechte: *Administrator*




**Safety Timeout**

Tragen Sie hier ein, nach welcher Zeit ohne netzwerkseitigen Zugriff die Outputs auf das eingestellte Muster gesetzt wer-

den sollen.

Der eingetragene Wert mal 100ms ergibt die Timeout-Zeit; es können Werte zwischen 1 und 9999 eingetragen werden.

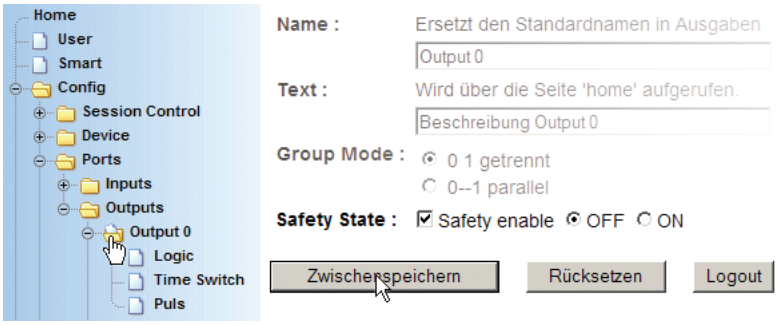
Der Wert 5 entspricht zum Beispiel 5 Sekunden.

Sinnvoll ist eine Timeout-Zeit, die ca. 10 mal so hoch ist, wie die Zeit zwischen zwei planmäßigen Zugriffen.

Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

### 6.2.2 Safety State festlegen

Wählen Sie im Navigationsbaum *Config >> Ports >> Outputs >>* und zum Beispiel *Output 0*




**Safety State**
Markieren Sie *Safety State Enable*, wenn der Output im Störungsfall auf eine feste Lage gesetzt werden soll. Legen Sie fest, ob der Output in Zustand *ON* oder *OFF* geschaltet werden soll.

Outputs die nicht markiert wurden, behalten den Zustand, der bei Eintreten der Störung anlag.

Alle weiteren Einstellungen auf dieser Seite sind für den *Safety State* nicht relevant und müssen nicht geändert werden.

Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Wenn alle Eingaben gemacht wurden, aktivieren Sie die neuen Einstellungen über *Config >> Session Control >> LogOut, Speichern.*

## 6.3 Ports - Input- und Output-Konfiguration

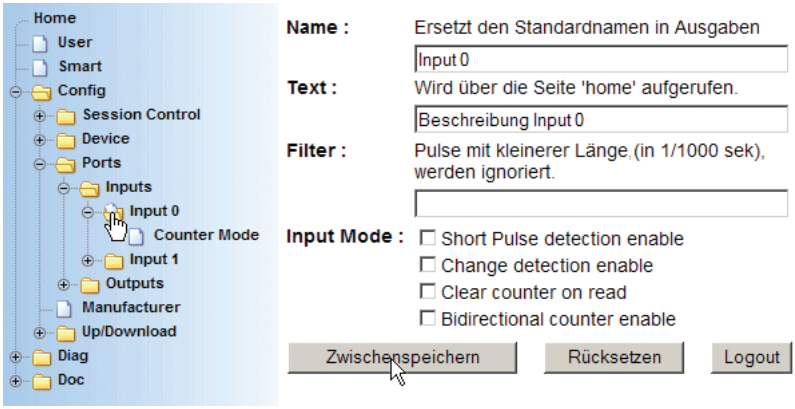
### 6.3.1 Input-Konfiguration

Die Inputs sind je nach Web-IO Modell in Gruppen bis zu vier Inputs zusammengefasst. Neben der galvanischen Trennung der einzelnen Gruppen voneinander, gibt es bei Gruppe 1 noch die Besonderheit, dass die Inputs einzeln zwischen Flanken- und Pulserkennung umgeschaltet werden können.

Für jeden Input können individuelle Grundeinstellungen vorgenommen werden.

Um z.B. die Einstellungen für Input 0 zu ändern, wählen Sie im Navigationsbaum *Config >> Ports >> Inputs* und für Input 0 *>> Input 0*:

Notwendige Zugriffsrechte: *Administrator* oder *Config*




**Name**
Geben Sie in dieses Feld eine Bezeichnung für den Input ein. Diese Bezeichnung wird dann im Browser für den Input 0 angezeigt.

**Text**
Hier kann eine nähere Beschreibung, z.B. zur Funktion des Inputs eingegeben werden, die im Browser durch Klick auf den Input-Link abgerufen werden kann.

**Filter**
Pulse mit kleinerer Länge, als der hier angegebene (in 1/1000 sek), werden ignoriert.

*Bei Inputs, die mit Short Puls Detection arbeiten, bleiben die eingestellten Filter ohne Wirkung.*

**Input Mode: Short Puls Detection Enable**
Dieser Modus ist nur für die ersten vier Inputs verfügbar und erlaubt auch die Erkennung von extrem kurzen Pulsen bis minimal 50µsLänge. In diesem Modus können nur komplette Pulse, aber keine Flankenwechsel gezählt werden.

Ist *Puls Detection* nicht aktiviert, müssen Signale für min. 1ms anliegen, damit sie erkannt und gezählt werden.

In beiden Fällen können max. 500 Zählungen pro Sek. erfasst werden.

**Input Mode: Change Detection Enable**
Neben der Zählung kompletter Pulse kann hier auf die Zählung von Flankenwechseln umgeschaltet werden.

Ist ohne Funktion, wenn *Short Puls Detection* enable aktiviert ist.

**Input Mode: Clear counter on read**
Hier kann ausgewählt werden, ob die Counter beim Auslesen automatisch auf 0 gesetzt werden sollen.

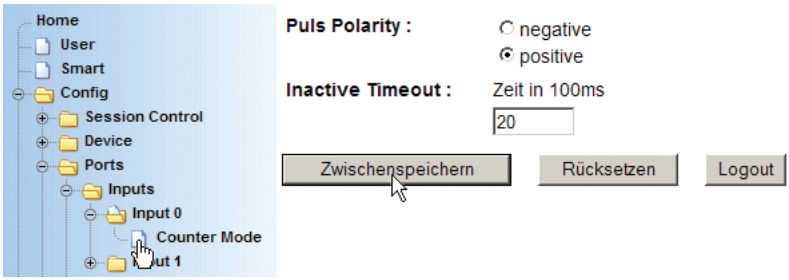
**Input Mode: Bidirectional counter enable**
Zwei benachbarte Inputs werden zur Auswertung von richtungsabhängigen Signalgebern zusammengefasst.

Je nachdem auf welchem der beiden Inputs zuerst eine gültige Flanke erkannt wird, wird der entsprechende Counter um eins hochgezählt.

Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Die nötigen Einstellungen für richtungsabhängiges Zählen werden unter *Config >> Ports >> Inputs >> Input0 >> Counter Mode* vorgenommen.

Notwendige Zugriffsrechte: *Administrator*




**Puls Polarity**
Wählen Sie hier, ob das Web-IO auf positive oder negative Flanken triggern soll.

**Inactivity Timeout**
Werden innerhalb der eingestellten Zeit nicht auf beiden Inputs gültige Flanken erkannt, wird die Zählung verworfen.

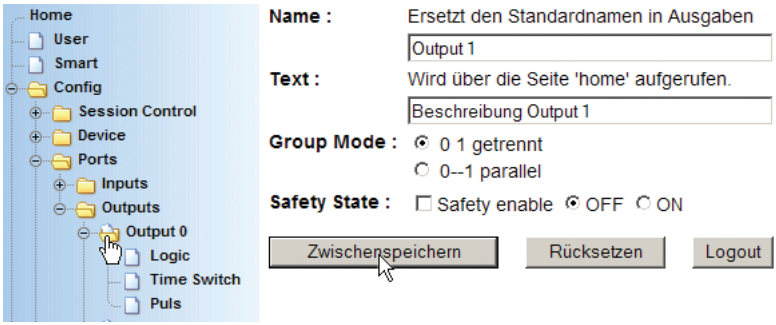
Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Wenn alle Eingaben gemacht wurden, aktivieren Sie die neuen Einstellungen über *Config >> Session Control >> LogOut, Speichern.*

### 6.3.2 Output-Konfiguration

Um die Einstellungen für z.B. Output 0 zu ändern, wählen Sie *Config >> Ports >> Outputs* und z.B. *>> Output 0:*

Notwendige Zugriffsrechte: *Administrator* oder *Config*




**Name**

Geben Sie in dieses Feld eine Bezeichnung für den Output ein. Diese Bezeichnung wird dann im Browser für den Output 0 angezeigt.

**Text**

Hier kann eine nähere Beschreibung z.B. zur Funktion des Outputs eingegeben werden, die im Browser durch Klick auf den Output-Link abgerufen werden kann.

**Group Mode**

*(nicht bei Web-IO mit Relaisausgängen)*

Die Outputs können zu Gruppen von zwei oder vier zusammengefasst werden, um den maximalen Schaltstrom von 500mA zu erhöhen. Durch Zusammenfassung von zwei Outputs kann ein maximaler Strom von 1A geschaltet werden. Bei vier Outputs sind es sogar 2A.

**Safety State**

Markieren Sie *Safety State Enable*, wenn der Output im Störungsfall auf eine feste Lage gesetzt werden soll. Legen Sie fest, ob der Output in Zustand *ON* oder *OFF* geschaltet werden soll. Unter *Config >> Device >> Output Mode* muss ein *Safety Timeout* eingetragen sein, damit der Safety State arbeitet.

Eine detaillierte Beschreibung zum *Safety State* finden Sie im Kapitel *Safety Timeout / State.*

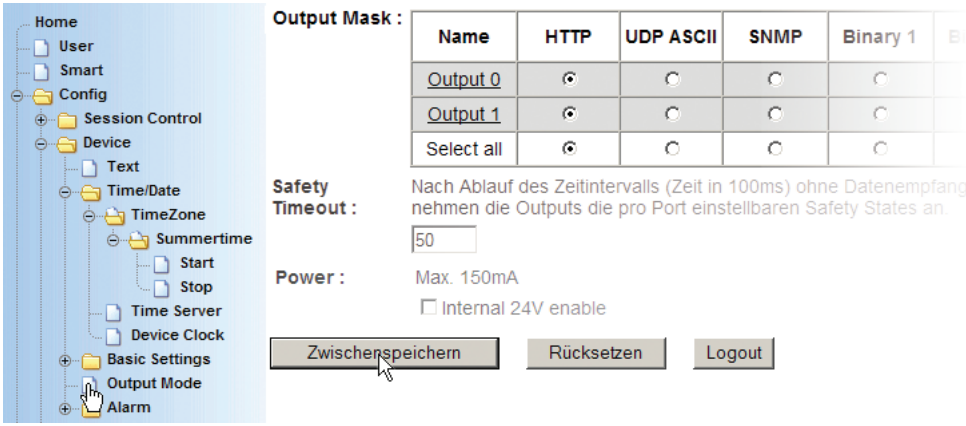
Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Wenn alle den Output betreffenden Eingaben gemacht wurden, aktivieren Sie die neuen Einstellungen über *Config >> Session Control >> Logout, Speichern.*

*Outputs, die als Gruppe zusammengefasst werden, sollten in der gleichen Betriebsart konfiguriert sein. Es kann sonst zu Konflikten kommen, wenn z.B. die Outputs 0 und 1 Box-to-Box arbeiten und die Outputs 2 und 3 mit dem Browser geschaltet werden können, da jeder Zugriff alle Outputs einer Gruppe auf einmal schaltet.*

Einen Überblick über die konfigurierten Output-Gruppen können Sie sich unter *Config >> Device >> Output Mode* verschaffen.

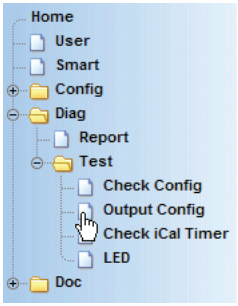
Notwendige Zugriffsrechte: *Administrator*




Hier sind die Outputs, die in einer Gruppe zusammengefasst wurden, in der gleichen Farbe hinterlegt.

Eine weitere Möglichkeit, die Gruppenbildung auf einen Blick anzusehen, finden Sie unter *Diag >> Test >> Output Config* in der Spalte *Group Mode.*

Notwendige Zugriffsrechte: *Administrator*




**Output Konfiguration und Status**

Von der Klemme direkt aufs Netzwerk

| Output | Modus | Group Mode | Safety State | Load Control |
|---|---|---|---|---|
| Output 0 | Binary 1 | 0--1 parallel | unverändert | inactive |
| Output 1 | Binary 1 | 0--1 parallel | unverändert | inactive |

last update: Mo, KW09, 28.02.2011 16:52:30

Binary 1 ModBus TCP Server waiting for TCP Client ...

| Output Power State | 0 mA |
|---|---|

### 6.3.3 Logik Funktionen

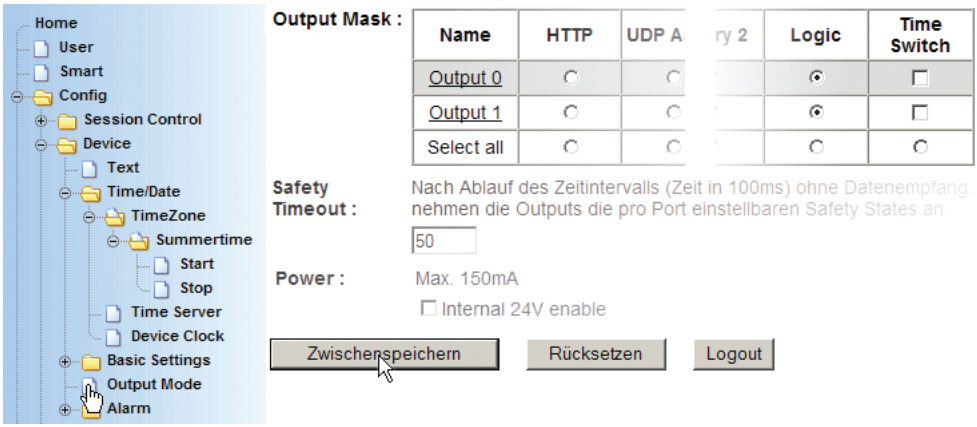
Das Web-IO erlaubt es, Outputs in Abhängigkeit des Zustands bestimmter Inputs zu schalten.

**Ein Beispiel**
Output 0 soll auf ON gesetzt werden, wenn Input 0 und Input 1 auf OFF stehen.

Damit die Logikfunktionen auf einen Output angewendet werden können, muss dieser zunächst auf die Betriebsart *Logic* konfiguriert werden.

Wählen Sie hierzu im Navigationsbaum *Config >> Device >> Outputmode.*

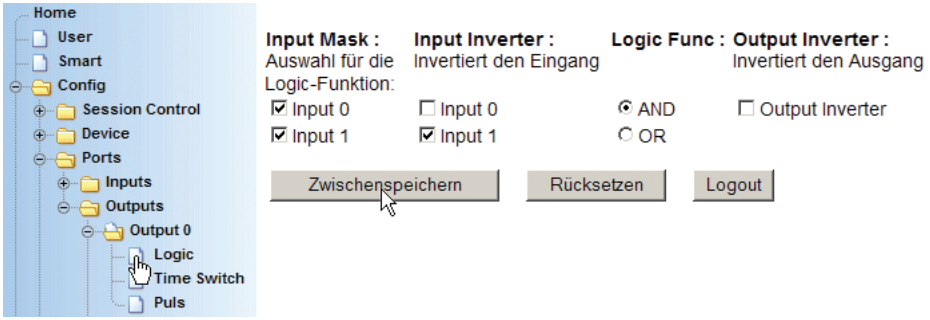
Notwendige Zugriffsrechte: *Administrator*




Markieren Sie für die Outputs, die über die Logikfunktion gesteuert werden sollen die Spalte *Logic* und klicken Sie auf *Zwischenspeichern*.

Wählen Sie nun im Navigationsbaum *Config >> Ports >> Outputs >> Output 0 >> Logic.*

Notwendige Zugriffsrechte: *Administrator*




**Beispiel**

Die in der Abbildung gezeigte Konfiguration setzt Output 0 auf ON, wenn Input 0 auf ON und Input 1 auf OFF gesetzt sind.

**Input Mask**

Markieren Sie hier alle Inputs, die bei der logischen Verknüpfung berücksichtigt werden sollen.

**Input Inverter**
Hier wird festgelegt, ob der ON oder OFF-Zustand des Eingangs berücksichtigt werden soll.

☐ = Input wird berücksichtigt wenn ON-Signal anliegt
☑ = Input wird berücksichtigt wenn OFF-Signal anliegt

**Logic Func**
Legt fest ob eine UND bzw. ODER-Verknüpfung angewendet wird.

**Output Inverter**
Hier wird festgelegt, ob bei Eintreten der logischen Bedingungen der Output auf ON oder auf OFF gesetzt werden soll.

☐ = Output wird auf ON gesetzt
☑ = Output wird auf OFF gesetzt

Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Wenn die Bedingungen für alle Outputs eingegeben wurden, aktivieren Sie die neuen Einstellungen über *Config >> Session Control >> LogOut, Speichern.*

### 6.3.4 Time Switch

Details zum zeitgesteuerten Schalten finden Sie im Kapitel „5.3 Time Switch - Web-IO als Schaltuhr“

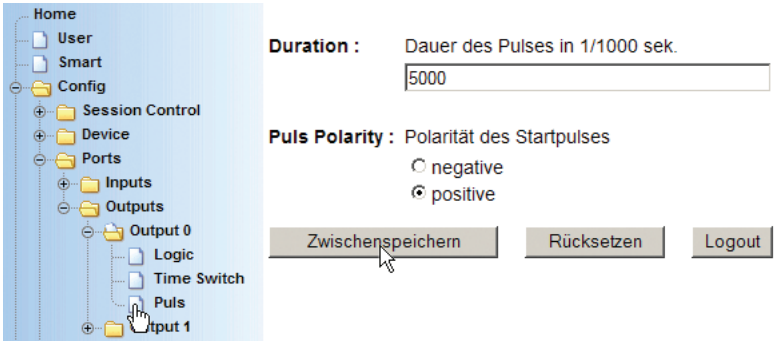
### 6.3.5 Pulsmode

Neben dem rein statischen Schalten der Outputs auf ON oder OFF erlaubt das Web-IO auch die Ausgabe von Pulsen. Das bedeutet, ein Output kann für eine voreinstellbare Zeit auf ON oder auf OFF geschaltet werden und fällt nach der eingestellten Pulslänge wieder zurück auf seinen Ruhezustand.

Um einen Output des Web-IO auf die Ausgabe von Pulsen

zu konfigurieren, wählen Sie im Navigationsbaum Config >> Ports >> Outputs >> Output0 >> Puls

Notwendige Zugriffsrechte: *Administrator*



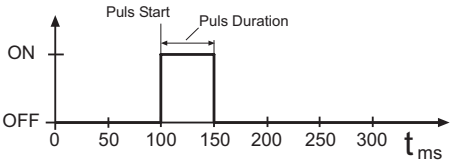

**Duration**

Tragen Sie hier die gewünschte Pulslänge in ms ein. Ein Wert von 1000 entspricht einem 1 Sekunde langen Puls.

Normalerweise kann ein Output während des Pulses nicht per Anwendereingriff zurück in den Ruhezustand geschaltet werden. Sollte das doch gewünscht sein, kann hinter der Pulslänge der Großbuchstabe *R* eingegeben werden (z.B. 2000R)

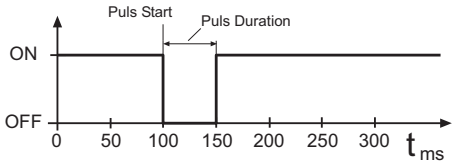
**Puls Polarity**

positiv:




Im Ruhezustand ist der Output nicht geschaltet (0V). Wird der Output nun per Browser oder aus einer Anwendung heraus auf ON gesetzt, schaltet das Web-IO für die eingestellte Pulsdauer die Versorgungsspannung +Vdd auf den Output.

negativ:




Im Ruhezustand ist der Output auf Versorgungsspannung +Vdd geschaltet. Wird der Output nun aus einer Anwendung heraus auf ON gesetzt, schaltet das Web-IO für die eingestellte Pulsdauer den Output ab (0V).

Wird bei *Puls Polarity negativ* als Pulslänge *0* eingegeben, führt das zu einer grundsätzlichen Invertierung des physikalischen Output-Signals.

*Es ist nicht möglich, einen negativen Puls via HTTP aus dem Browser zu schalten. Wenn aus dem Browser mit negativen Pulsen gearbeitet werden soll, kann das nur aus eigenen Anwendungen über HTTP-Requests oder das Socket-Interface erfolgen*

*Bei negativer Polarität wird mit negativer Logik gearbeitet. Ein Schalten nach ON bewirkt ein Abschalten des entsprechenden Output.*

*Bei Web-IO mit Relaisausgängen, muss die Pulslänge min. 200ms betragen!*

Über *Logout* und *Speichern* werden die neuen Einstellungen aktiviert.

## 6.4 Ändern der gerätespezifischen Anzeigen

Für projektbezogene OEM-Anwendungen des Web-IO ist es möglich, Anzeigen wie Artikelnummer, Hersteller, Logo usw. zu verändern.

Wählen Sie im Navigationsbaum *Config >> Manufacturer:*

Notwendige Zugriffsrechte: *Administrator*

**Name**
Feld für den Firmennamen

**Address**
Feld für die Firmenanschrift

**Support-Hotline**
Rufnummer, unter der der Anwender technische Hilfe zum Web-IO bzw. zum Gesamtprojekt bekommen kann

**Internet**
Hompage der für das Gerät zuständigen Firma

**Device Type**
Firmenspezifische Gerätebezeichnung

**Device No**
Firmenspezifische Artikelnummer

**Logo**
Pfad und Name des über dem Navigationsbaum positionierten Logos. Wird nur logo.gif angegeben, greift das Web-IO auf das intern abgelegte Logo zurück. Alternativ kann aber auch eine JPG- oder GIF-Datei auf einem anderen Server angegeben werden.

Dabei sollte allerdings beachtet werden, dass die Abmessungen des Logos in etwa mit denen des W&T-Logos übereinstimmen (Breite max. 200 Pixel, Höhe max. 100 Pixel), damit die Übersichtlichkeit des Navigations-Frames erhalten bleibt.

Übergeben Sie die Einstellungen durch Klick auf den *Zwischenspeichern*-Button an das Web-IO.

Aktivieren Sie die neuen Einstellungen über *Config >> Session Control >> LogOut, Speichern.*

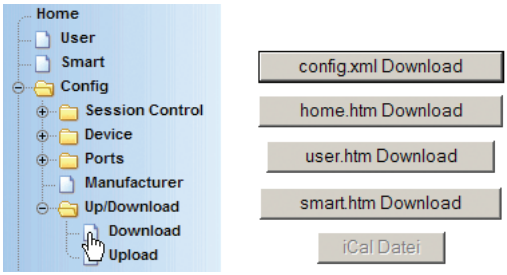
## 6.5 Upload/Download

Das Web-IO Digital bietet dem Anwender die Möglichkeit, kundenspezifische HTML-Seiten, Konfigurationsdaten und das Logo per Upload einzuspielen bzw. per Download herunterzuladen.

### 6.5.1 Download

Um eine HTML-Seite oder die Konfiguration herunterzuladen wählen Sie im Navigationsbaum *Config >> Up/Download >> Download.*

Notwendige Zugriffsrechte: *Administrator*




**Config.xml Download**
Die gesamte Konfiguration des Web-IO ist im Gerät als XML-Datei abrufbar. So kann der Administrator bei Projekten, in denen mehrere Web-IO mit gleicher Konfiguration zum Einsatz kommen, bei einem Gerät die Grundkonfiguration vornehmen, auslesen und später in die anderen einspielen.

Starten Sie den Download der XML-Datei durch Klick auf den entsprechenden Button.

Die XML-Parameter werden daraufhin im Browserfenster angezeigt. Um die XML-Daten als File abzuspeichern kann bei den meisten Browsern über die rechte Mouse-Taste der Quelltext im Texteditor angezeigt und von dort aus natürlich auch gespeichert werden.

**home.htm / user.htm / smart.htm Download**
Auch hier wird die gewählte Seite zunächst im Browserfenster angezeigt und kann über den Weg der Quelltextanzeige abgespeichert werden. Eine detaillierte Beschreibung hierzu finden Sie im Kapitel *HTTP*.

**iCal Datei**
Sollte das Web-IO als Schaltuhr konfiguriert sein, können hier die derzeit aktiven iCal-Dateien zur Kontrolle heruntergeladen werden.

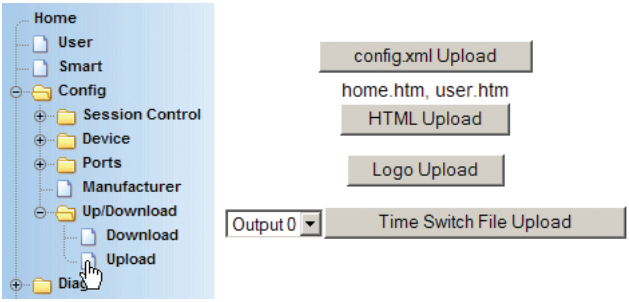
*Details zum zeitgesteuerten Schalten finden Sie im Kapitel „5.3 Time Switch - Web-IO als Schaltuhr“ .*

### 6.5.2 Upload

So wie HTML- und XML- Dateien heruntergeladen werden können, ist es auch möglich, diese wieder in das Web-IO einzuspielen.

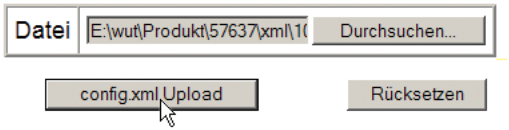
Wählen Sie hierzu im Navigationsbaum *Config >> Up/Download >> Upload:*

Notwendige Zugriffsrechte: *Administrator*




**config.xml Upload**
Die gesamte Konfiguration des Web-IO kann als XML-Datei in einem Zug in das Web-IO eingespielt werden. Nach Klick auf den entsprechenden Button, wird im Konfigurations-Frame ein Eingabefeld angezeigt, in dem das Upload-File eingetra-

gen bzw. ausgewählt werden kann:




Durch Klick auf den *Upload*-Button, wird das Upload gestartet.

Eine komfortable Möglichkeit XML-Dateien aus dem Web-IO zu speichern, zu verwalten und wieder einzuspielen bietet das Wutility-Tool.

*Auf unserer Webseite www.WuT.de finden Sie links die„Suche über Artikel-Nr". Geben Sie hier die Artikel-Nr. Ihres Web-IO ein, wählen Sie in dem Feld darunter „Tools" aus und klicken Sie auf „Los". Auf der Seite, die sich dann öffnet, wählen Sie den Link zum Wutility-Tool.*

**HTML Upload**

Der Update-Vorgang für die home.htm, user.htm und Seite läuft genauso ab, wie das XML-Upload.

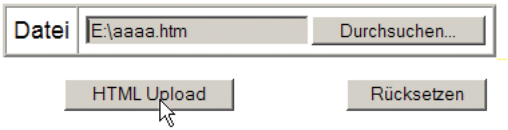
Allerdings benötigt das Web-IO für die korrekte Zuordnung der Datei in der ersten Zeile des HTML-Quelltextes das <home.htm>, <user.htm> bzw. <smart.htm> Tag steht.

Die genaue Vorgehensweise ist im Kapitel *Zugriff aus der eigenen Anwendung >> Zugriff aus dem Browser* detailliert beschreiben.

**logo Upload**

Um ein möglichst kundenspezifisches Erscheinungsbild der Web-IO Startseite zu ermöglichen, besteht die Möglichkeit, das W&T-Logo im Navigations-Frame gegen ein beliebiges anderes zu ersetzen.

Nach Klick auf den entsprechenden Button, wird im Konfigurations-Frame ein Eingabefeld angezeigt, in dem das Upload-File eingetragen bzw. ausgewählt werden kann:




Durch Klick auf den Upload-Button, wird das Upload gestartet.

**Time Switch File Upload**

Hier können Kalenderdateien im iCal-Format hochgeladen werden, die das zeitgesteuerte Schalten bestimmen.

*Details zum zeitgesteuerten Schalten finden Sie im Kapitel „5.3 Time Switch - Web-IO als Schaltuhr“*

Damit die Uploads auch nach Unterbrechung der Stromversorgung noch Wirkung haben, aktivieren Sie die neuen Einstellungen über *Config >> Session Control >> LogOut, Speichern.*

## 6.6 Diagnose und Test

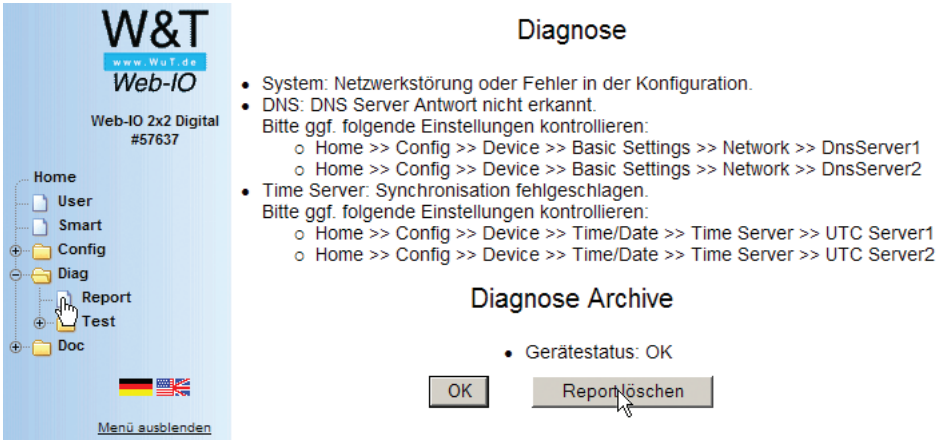
Das Web-IO verfügt über ein internes Fehlermanagement und Diagnosesystem.

### 6.6.1 Fehler-Report

Tritt ein Fehler auf, wird dies am Gerät selber durch Blinken der *on error* Led angezeigt. Darüber hinaus werden aufgetretene Fehlerzustände in einem Report dokumentiert und können später ausgelesen werden.

Alle Fehlermeldungen werden im Web-IO gespeichert und bleiben auch erhalten, wenn die Fehlerursache bereits behoben ist.

Wählen Sie dazu im Navigationsbaum *Diag >> Report:*




Durch Klick auf den *Report löschen* Button kann der Administrator alle Fehlermeldungen löschen.

Notwendige Zugriffsrechte: *Administrator*

Ein Reset, unabhängig ob er durch Drücken der Resettaste am Gerät, durch Unterbrechung der Versorgungsspannung

oder durch Reset aus der Logout Seite ausgelöst wurde, löscht ebenfalls den Report.

*Eine genaue Liste möglicher Fehlermeldungen und deren Ursache finden Sie im Anhang. Im Kapitel Socket-Programmierung wird gezeigt, wie Fehler auch aus eigenen Anwendungen heraus verarbeitet werden können.*

*Darüber hinaus können Fehler-und Diagnosemeldungen auch über SNMP-Traps oder als Syslog-Systemmeldung verarbeitet werden. Siehe hierzu die Kapitel SNMP bzw. Syslog*

### 6.6.2 Konfiguration

**Gesamtüberblick**

Das Web-IO bietet dem Administrator die Möglichkeit, die Konfiguration des Web-IO auf einer übersichtlichen Webseite zu überblicken und zu überprüfen.

Wählen Sie im Navigationsbaum *Diag >> Test >> Check Config.*

**Prüfung der Konfiguration**

| Parameter | HTTP | UDP | SNMP | Mail | Syslog | FTP | Binary 1 | Binary 2 | Logic | Time Switch |
|---|---|---|---|---|---|---|---|---|---|---|
| Enable Flag | ---- | OFF | ON | OFF | OFF | OFF | ON | OFF | ---- | ---- |
| Output Mode | OK | OFF | OFF | ---- | ---- | ---- | OFF | OFF | OFF | OFF |
| Source Port | 80 | 42279 | 161 | auto | auto | auto | ---- | 49154 | ---- | ---- |
| Source IpAddr | 10.40.22.126 | 10.40.22.126 | 10.40.22.126 | 10.40.22.126 | 10.40.22.126 | 10.40.22.126 | 10.40.22.126 | 10.40.22.126 | ---- | ---- |
| Destination Port | n.a. | n.a. | ---- | ---- | ---- | ---- | ---- | ---- | n.a. | n.a. |
| Destination IpAddr | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- |
| Logic Funcs | ---- | ---- | ---- | ---- | ---- | ---- | ---- | ---- | OFF | ---- |
| Active | **ON** | OFF | **ON** | OFF | OFF | OFF | FAIL | OFF | OFF | OFF |
| Profile | Browser, ASCII Kommandostrings | ASCII Kommandostrings per UDP | SNMP | Mail | Syslog | FTP | Box-2-Box, Binärer Socketzugriff, OPC | | Logik | Time Switch |

Fehlerhafte oder unvollständige Eingaben werden orange markiert. Wählen Sie in diesem Fall unter Config>>Session Control>>Profiles das entsprechende Profil aus und überprüfen Sie die blau gekennzeichneten Parameter.

| Parameter | Alarm Mail | SNMP Trap | UDP Client | Pending TCP | TCP Client | Syslog Message | FTP Message |
|---|---|---|---|---|---|---|---|
| Alarm / Trap | OFF | OFF | OFF | OFF | OFF | OFF | OFF |

| Connection Recources | |
|---|---|
| Mode | Max |
| Pending TCP | 3 |

Die Webseite zeigt, welche Zugriffsarten mit welchen Para-

metern aktiviert sind. Dabei nimmt das Web-IO eine Plausibilitätsprüfung der Einstellungen vor. Werden fehlende Parameter erkannt, die den ordnungsgemäßen Betrieb der Zugriffsart verhindern, werden die entsprechenden Felder orange hinterlegt. Ein Klick auf den Fail-Link führt direkt zu der fehlerhaften Konfiguration.

Ferner wird überprüft und angezeigt, welche Übertragungswege für die Alarme gewählt wurden und ob alle benötigten Parameter konfiguriert wurden. Auch hier werden die Zugangsarten orange hinterlegt, die nicht vollständig konfiguriert wurden.

Zuletzt wird überprüft und angezeigt wieviele statische Verbindungen auf Port 80 des Web-IO möglich sind. Der Port 80 des Web-IO kann sowohl für den Aufruf der Web-Seiten, als auch für Socketanwendungen genutzt werden.

Bei Werkseinstellungen sind alle freien Ports so konfiguriert, dass sie auf Port 80 als Serverport arbeiten. Dabei werden auf zwei Ports nur dynamische Verbindungen zugelassen.

Sind z.B. 5 Ports auf HTTP (Port 80 ) konfiguriert, können 3 davon durch dauerhafte TCP-Verbindungen, z.B. von Socket-Anwendungen genutzt werden. Bei den verbleibenden zwei freien Ports werden Verbindungen zwar entgegengenommen, vom Web-IO aber nach kurzer Timeout-Zeit wieder abgeworfen.

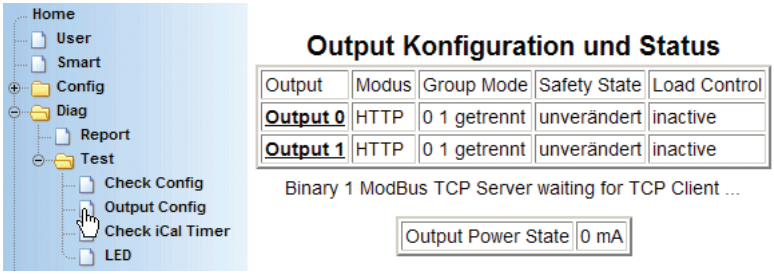
Durch diese Maßnahme ist gewährleistet, dass man das Web-IO zu jeder Zeit über den Browser erreichen kann.

**Überblick der Output-Konfiguration**

Neben der Übersicht über die allgemeine Konfiguration bietet das Web-IO noch eine zusätzliche Webseite, die die Konfiguration und den Status der Outputs wiedergibt.

Wählen Sie im Navigationsbaum *Diag >> Test >> Output Config*

⌨ Notwendige Zugriffsrechte: *Administrator*




Für jeden der Outputs werden folgende Informationen ausgegeben:

**Modus**
Hier wird die konfigurierte Betriebsart angezeigt:

- HTTP
- UDP ASCII
- SNMP
- Binary 1
- Binary 2
- Logic

**Group Mode**
Wurden 2 oder 4 Outputs zu einer Gruppe zusammengefasst, wird das in dieser Spalte angezeigt.

**Safety State**
Diese Spalte kann folgende Einträge enthalten:

- *unverändert*
  Es wurde keine Watchdog Funktion für diesen Ausgang gesetzt.
- *ON*
  Es wurde eine Watchdog Funktion für diesen Ausgang festgelegt.
- *Error*
  Bei aktivierter Load Control wurde eine Überlast oder eine fehlende Last erkannt.
  Bei Überlast wurde der Output vom Web-IO abgeschaltet.
  *(nicht bei Web-IO mit Relaisausgängen)*

*Da es sich im letzten Fall um einen schwerwiegenden Fehler handelt, werden die Felder im Fehlerfall rot hinterlegt.*

**Load Control**

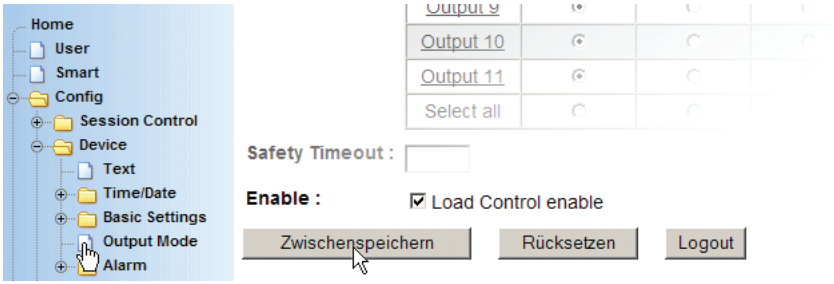
*(nicht bei Web-IO mit Relaisausgängen und PoE versorgten Geräten)*

Das Web-IO Digital verfügt neben der thermischen Kurzschlusssicherung über eine Lastüberwachung, die im Fehlerfall die Outputs solange abschaltet, bis sie vom Administrator wieder freigeschaltet werden.

In den Werkseinstellungen ist die Load Control nicht aktiviert.

Unter *Config >> Device >> Output Mode* kann durch das Markieren von *Load Control Enable* die Lastüberwachung aktiviert bzw. deaktiviert werden.

Notwendige Zugriffsrechte: *Administrator*




*Unabhängig davon, ob Load Control aktiviert ist oder nicht, schützt die thermische Kurzschlusssicherung den betroffenen Output vor Beschädigung.*

Mit der Aktivierung von Load Control wird ein Management eingeschaltet, welches im Überlastfall die betroffenen Outputs in den Status OFF schaltet und den Anwender informieren kann.

Aus hardwaretechnischen Gründen findet die Lastüberwachung paarweise, also immer für zwei Outputs statt, z.B. Output 0 und Output 1. Je nach Zustand der Outputs und Art des Fehlers kann das Web-IO nicht eindeutig bestimmen,

welcher Output ursächlich für den Fehler ist. In diesem Fall werden beide Outputs auf OFF gesetzt.

Es wird zusätzlich überwacht, ob überhaupt eine Last vorhanden ist. Kann das Web-IO eindeutig feststellen, dass es sich bei dem aufgetretenen Fehler um eine fehlende Last handelt, wird (nur) für dieses Outputpaar die Load Control deaktiviert.

Durch Klick auf den *Report löschen* Button unter *Diag >>Report* kann der Administrator abgeschaltete Outputs wieder einschalten.

Notwendige Zugriffsrechte: *Administrator*

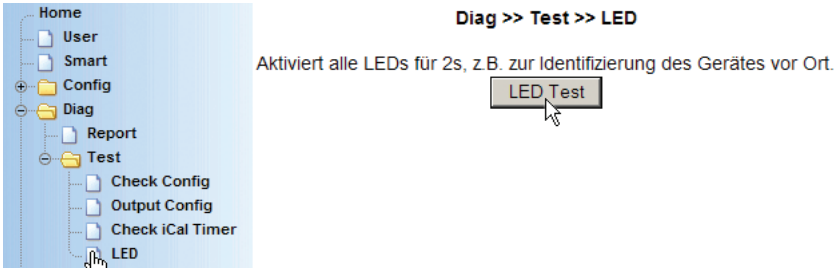
*Bei Fehlermeldungen durch Load control muss in jedem Fall die physikalische Beschaltung der betroffenen Outputs überprüft werden, bevor diese wieder freigeschaltet werden.*

*Overload-Fehler können auch als Trigger für einen Alarm konfiguriert werden. Siehe Kapitel* Alarme

### 6.6.3 LED -Test

Um alle LEDs des Input/Output-Teils des Web-IO Digital auf Funktion zu prüfen wählen Sie im Navigationsbaum *Diag >> Test >> LED* und klicken Sie auf den *LED Test* Button.

Notwendige Zugriffsrechte: *Default User, Config, Administrator.*




Alle LEDs werden für ca. 2 Sek. eingeschaltet.

Mit Hilfe dieser Funktion kann ein Web-IO im Schaltschrank eindeutig identifiziert werden.

*Am tatsächlichen Zustand der Outputs ändert sich natürlich nichts. Die 3 LEDs über dem Ethernet-Anschluss werden von diesem Test nicht berührt.*

# 7 Anhang

- Alternative bei der IP-Adressvergabe
- Firmware Update
- Notzugang
- Verwendete Portnummern und Netzwerksicherheit
- Fehlermeldungen und ihre Ursachen
- Technische Daten
- Konformitätserklärung
- Index

## 7.1 Alternativen bei der IP-Adressvergabe

In den meisten Fällen in denen die IP-Adressvergabe nicht per DHCP erfolgt ist es am einfachsten, dem Web-IO eine IP-Adresse mit dem Wutility Tool zuzuweisen (Siehe Abschnitt 2.2).

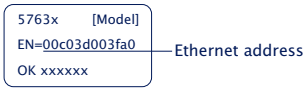
In Ausnahmefällen oder wenn kein PC mit Windows-System zur Verfügung steht bieten sich folgende Alternativen:

- Vergabe mit Hilfe des ARP-Kommandos
- Vergabe über die serielle Schnittstelle
- Vergabe per RARP-Protokoll
- Vergabe per DHCP-Protoll
- Vergabe per BOOTP-Protoll

### 7.1.1 Vergabe der IP mit Hilfe des ARP-Kommandos

*Diese Methode ist nur ausführbar, wenn das Web-IO noch keine IP-Adresse hat, der Eintrag also 0.0.0.0 lautet. Verwenden Sie zum Ändern einer IP-Adresse eine der anderen in diesem Kapitel beschriebenen Methoden oder den Menüpunkt Config/Device/Network im Web-Based Management des Web-IO.*

Weitere Voraussetzung ist ein Computer, der sich im gleichen Netzwerksegment des Web-IO befindet und auf dem das TCP/IP-Protokoll installiert ist. Lesen Sie die Ethernet-Adresse des Web-IO von dem Aufkleber an der Gehäuseseite ab:




Fügen Sie jetzt mit der folgenden Befehlszeile der ARP-Tabelle des Rechners einen statischen Eintrag hinzu:

```
arp -s [IP-Adresse]  [MAC-Adresse]
```

Beispiel unter Windows:

```
arp -s 10.40.72.15 00-C0-3-00-3F-A0
```

Beispiel unter SCO UNIX:

```
arp -s 10.40.72.15 00:C0:3D:00:3F:A0
```

*Ältere Windows-Systeme akzeptieren einen statischen Eintrag nur dann, wenn bereits ein dynamischer vorhanden ist. Führen Sie hier zunächst ein PING auf einen anderen Netzwerkteilnehmer durch.*

Starten Sie abschließend den Web-Browser und geben Sie folgende URL ein:

http://[IP-Adresse] [*Return*]

*Beispiel:*

*In Windows-Umgebungen darf die Eingabe von IP-Adressen nur ohne führende Nullen erfolgen. Ansonsten wird die Eingabe vom System falsch interpretiert und dem Web-IO wird eine falsche IP-Adresse zugewiesen.*

Das Web-IO übernimmt die IP-Adresse des ersten an seine Ethernet-Adresse gesendeten Netzwerkpaketes als seine eigene und speichert diese nichtflüchtig ab. Die HTTP-Verbindung wird daraufhin aufgebaut und die Homepage des Web-IO erscheint im Browser. Alle weiteren Einstellungen können nun bequem per Web-Based Management vorgenommen werden (siehe *Kapitel Grundeinstellung der Netzwekparameter*).

### 7.1.2 Vergabe der IP über die serielle Schnittstelle

Nach einem Reset des Web-IO wird am seriellen Port ein Zeitfenster von ca. 1-2 Sekunden zur Verfügung gestellt, in dem durch die Eingabe von mindestens 3 „x“ die Vergabe einer neuen IP-Adresse ermöglicht wird.

*Im Gegensatz zur vorher beschriebenen Methode über ARP funktioniert der serielle Weg unabhängig davon, ob das Web-IO bereits eine IP-Adresse hat oder nicht. Der*

*Vorgang ist beliebig oft wiederholbar. Verwenden Sie deshalb diese Methode, wenn Sie die IP-Adresse nicht kennen oder vergessen haben.*

*Der Anhang enthält hierzu eine detaillierte Vorgehensweise unter Windows mit dem Terminalprogramm Hyperterminal.*

Verbinden Sie zunächst das Web-IO seriell mit einem Computer. Für einen Standard-PC wird ein *gekreuztes* RS232-Kabel (=Nullmodemkabel) benötigt (siehe Kapitel *Serieller Anschluss*).

Die seriellen Übertragungsparameter des verwendeten Terminalprogramms werden auf *9600 Baud, no Parity, 8 Bits, 1 Stopbit, no Handshake* konfiguriert. Betätigen Sie kurz die Resettaste des Web-IO und geben Sie am Terminal mindestens dreimal den Buchstaben „x“ ein bis im Terminalprogramm das Prompt *IPno.+<Enter>:* erscheint.

Geben Sie im üblichen Format (xxx.xxx.xxx.xxx) die IP Adresse ein, und beenden Sie die Eingabe mit <*Enter*>. Wurde die Eingabe akzeptiert, wird mit der zugewiesenen IP-Adresse quittiert. Ansonsten erfolgt die Meldung *FAIL* gefolgt von der zuletzt aktuellen IP-Adresse.

Alle weiteren Einstellungen können nun bequem per Web-Based Management im Browser vorgenommen werden (siehe Kapitel *Grundeinstellung der Netzwerkparameter*).

**Serielle Vergabe von Subnet-Mask und Gateway**

Zusammen mit der IP-Adresse können auch die Subnet-Mask und Gateway-Adresse seriell vergeben werden. Die Angabe erfolgt durch Kommata getrennt, im Anschluss an die IP-Adresse. Durch die Eingabe im folgenden Beispiel wird dem Web-IO die IP-Adresse 10.40.72.15, die Subnet-Mask 255.255.0.0 und das Gateway 10.40.1.1zugewiesen.

```
IP no.+<ENTER>:                              <- Web-IO
10.40.72.15,255.255.255.0,10.40.1.1          -> Web-IO
10.40.72.15 ,255.255.255.0,10.40.1.1-1       <- Web-IO
```

**Serielle Deaktivierung von DHCP-/BOOTP/RARP**
Die DHCP-/BOOTP-Funktion und RARP des Web-IO kann im Zuge einer seriellen Vergabe der IP-Adresse abgeschaltet werden. Wir empfehlen, hiervon in allen Fällen Gebrauch zu machen, in denen nicht ausdrücklich die IP-Zuweisung mittels DHCP/BOOTP oder RARP erfolgen soll.

Geben Sie zur Deaktivierung des DHCP-/BOOTP-Clients bzw. RARP im direkten Anschluss (ohne Leerzeichen!) an die IP-Adresse die Option „-0“ (null) an und beenden Sie die Eingabe mit <*Enter*>.

Beispiel:

```
xxx                                      -> Web-IO
IP no.+<ENTER>:                          <- Web-IO
10.40.72.15,255.255.255.0,10.40.1.1-0    -> Web-IO
10.40.72.15,255.255.255.0,10.40.1.1-0    <- Web-IO
```

Das funktioniert selbstverständlich genauso, wenn Subnet-Mask und Gateway mit eingegeben werden.

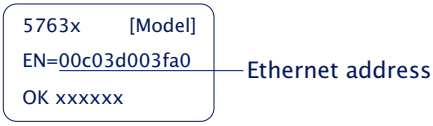
Die Funktion kann später jederzeit wieder über Web-Based Management im Browser aktiviert werden.

Menüpunkt: *Config >> Device >> Network*

### 7.1.3 IP-Vergabe über einen RARP-Server

Besonders UNIX-Umgebungen bedienen sich für eine zentralisierte Zuweisung von IP-Adressen häufig des RARP-Protokolls. Hierbei senden TCP/IP-Geräte, die eine IP-Adresse beziehen möchten, RARP-Requests mit ihrer Ethernet-Adresse als Broadcast in das Netzwerk.

Aktivieren Sie den RARP-Server auf dem UNIX-System, und tragen Sie in der Datei */etc/ethers* die Ethernet-Adresse des Web-IO ein. In der Datei */etc/hosts* muss zusätzlich die IP-Adresse des Web-IO eingetragen werden.




Das Web-IO muss sich im gleichen Subnetz wie der RARP-Server befinden.

**Beispiel**
Ihr Web-IO hat die MAC-Adresse EN= 00C03D003FA0 (Sticker auf dem Gerät). Er soll die IP-Adresse 172.16.232.17 und den Aliasnamen WEBIO_1 erhalten:

- Eintrag in der Datei */etc/hosts*:
  `10.40.72.15WEBIO_1`
- Eintrag in der Datei */etc/ethers*:
  `00:C0:3D:00:3F:A0 WEBIO_1`

Lautet die IP-Adresse des Web-IO 0.0.0.0 (=Factory Defaults), werden zyklisch RARP-Broadcasts erzeugt, um von einem ggf. vorhandenen RARP-Daemon eine gültige Adresse zu beziehen.

Verfügt das Web-IO bereits über eine gültige IP-Adresse, wird nach jedem Reset ein RARP-Broadcast erzeugt. Erfolgt hierauf innerhalb von 500ms ein Reply, wird die darin enthaltene IP-Adresse übernommen. Wie bei der Vergabe über die serielle Schnittstelle, ist es auf diesem Wege ebenfalls möglich, eine aktuelle IP-Adresse zu überschreiben.

### 7.1.4 IP-Vergabe per DHCP-Protokoll

Viele Netzwerke nutzen für die zentralisierte und dynamische Vergabe der Netzwerkparameter DHCP (**D**ynamic **H**ost **C**onfiguration **P**rotocol) oder auch das im folgenden Kapitel beschriebene Vorgängerprotokoll BOOTP. Mit den Werkseinstellungen ist das DHCP-Protokoll aktiviert, so dass es in Netzwerkumgebungen mit dynamischer IP-Vergabe ausreicht, das Web-IO an das Netzwerk anzuschließen. Die folgenden Parameter können mit Hilfe von DHCP zugewiesen werden:

- IP-Adresse
- Subnet-Mask
- Gateway-Adresse
- DNS-Server
- Lease-Time

*Zur Vermeidung ungewollter Adressvergaben oder Adressänderungen, empfehlen wir die Protokolle DHCP, BOOTP und RARP zu deaktivieren, sofern diese nicht ausdrücklich in der jeweiligen Netzwerkumgebung genutzt werden. Web-IO mit fälschlich zugeteilter IP-Adresse können nachträglich mit Hilfe der Scan-Funktion des Managementtools Wutility komfortabel gefunden und neu konfiguriert werden.*

**Aktivierung/Deaktivierung von DHCP**
Mit der Werkseinstellung ist das DHCP-Protokoll aktiviert. Zur Deaktivierung oder auch zum späteren Wiedereinschalten stehen die folgenden Möglichkeiten zur Verfügung.

- **Management-Tool WuTility**
  Markieren Sie in der Geräteliste das gewünschten Web-IO und betätigen den Button *IP-Adresse*. In dem ersten Dialogfenster tragen Sie die neu zu vergebenden Netzwerkparameter ein und betätigen dann *Weiter*. Deaktivieren Sie in dem folgenden Dialogfenster die Optionen *BOOTP* und *DHCP*. Mit *Weiter* werden jetzt abschließend die neuen Konfigurationsdaten an das Web-IO übertragen.

- **Serielle Schnittstelle**
  Im Zuge der seriellen IP-Vergabe, können in direktem Anschluss an den Adressstring folgende Optionen zur Deaktivierung/Aktivierung der Protokolle DHCP, BOOTP angegeben werden:
  **-0** → Deaktiviert DHCP und BOOTP
  **-1** → Aktiviert BOOTP/RARP
  **-2** → Aktiviert DHCP
  Eine detaillierte Beschreibung der Vorgehensweise finden Sie im Kapitel *Serielle Vergabe von IP, Subnet Mask und Gateway*.

- **Web Based Management**

Im Menüzweig *Config >> Device >> Basic Settings >> Network -> DHCP/BOOTP Enable* können die Protokolle alternierend aktiviert bzw. beide deaktiviert werden. Detailinformationen hierzu finden Sie im Kapitel *Konfiguration der Netzwerkparameter.*

**System Name**
Zur Unterstützung einer eventuell automatisierten Aktualisierung des DNS-Systems durch den DHCP-Server, identifiziert sich das Web-IO innerhalb des DHCP-Protokolls mit seinem System Namen. In der Werkseinstellung lautet dieser *WEBIO-* gefolgt von den letzten drei Stellen der Ethernet-Adresse. Zum Beispiel lautet der werksseitig eingestellte System Name (Device Name) eines Web-IO mit der Ethernet-Adresse 00:c0:3d:01:02:03 *WEBIO-010203*. Der Systemname des Web-IO kann per Web-Based Management geändert werden. Weitere Informationen hierzu finden Sie im Kapitel Menü: *Konfiguration der Netzwerkparameter.*

**Lease-Time**
Die vom DHCP-Server bestimmte und übermittelte Lease-Time legt die Gültigkeitsdauer der zugewiesenen IP-Adresse fest. Nach Ablauf der halben Lease-Time versucht das Web-IO bei dem zuweisenden DHCP-Server die Gültigkeit zu verlängern bzw. die Adresse zu aktualisieren. Ist dieses bis zum Ablauf der Lease-Time nicht möglich, zum Beispiel weil der DHCP-Server nicht mehr erreichbar ist, löscht das Web-IO die IP-Adresse und startet eine zyklische Suche nach alternativen DHCP-Servern zwecks Zuweisung einer neuen IP-Adresse.

Bedingt durch die fehlende Uhr, ist die zur aktuellen IP-Adresse gehörende Lease-Time nach einem Reset nicht mehr verfügbar. Nach dem Neustart erfolgt daher eine entsprechende Aktualisierungsanfrage bei dem ursprünglichen DHCP-Server. Sollte dieser zu diesem Zeitpunkt nicht erreichbar sein, löscht das Web-IO die IP-Adresse und startet eine zyklische Suche nach alternativen DHCP-Servern.

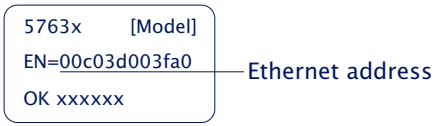
Ist DHCP aktiviert, wird die verbleibende Lease-Time zusammen mit der aktuellen IP-Adresse im Menüzweig *Config >>*

*Device >> Basic Settings >> Network* im Format hh:mm:ss angezeigt.

⚠ *Sollte nach Ablauf der zugewiesenen Lease-Time der DHCP-Server nicht erreichbar sein, löscht das Web-IO seine IP-Adresse.* Alle *bestehenden TCP/UDP-Verbindungen zwischen dem Web-IO und anderen Netzwerkteilnehmern werden hierdurch unterbrochen. Um Störungen dieser Art zu vermeiden, empfehlen wir, die zu vergebende Lease-Time im DHCP-Server möglichst auf* unendlich *zu konfigurieren.*

**Reservierte IP-Adressen**

Wird das Web-IO als TCP-Server oder UDP-Peer eingesetzt, stellt es Dienste zur Verfügung, die andere Teilnehmer (Clients) im Netzwerk nach Bedarf in Anspruch nehmen können. Für die Verbidungsaufnahme wird von diesen natürlich die aktuelle IP-Adresse des Web-IO benötigt, so dass es in diesen Anwendungsfällen sinnvoll ist, auf dem DHCP-Server eine bestimmte IP-Adresse für das Web-IO zu reservieren. In der Regel erfolgt dieses durch die Bindung der IP-Adresse an die weltweit einmalige Ethernet-Adresse des Web-IO, welche dem Aufkleber am Gehäuse entnommen werden kann.




**Dynamische IP-Adressen**

Eine völlig dynamische Adress-Vergabe, bei welcher das Web-IO mit jedem Neustart oder auch nach Ablauf der Lease-Zeit eine andere IP-Adresse bekommt, ist nur in Netzwerkumgebungen mit automatisierter Querverbindung zwischen den Diensten DHCP und DNS sinnvoll. Das heißt, bei der Neuzuteilung einer IP-Adresse an das Web-IO, aktualisiert der DHCP-Server anschließend automatisch auch das DNS-System. Dem jeweiligen Domain-Namen wird hierbei die neue Adresse zugeordnet. Für Detailinformationen zu Ihrer Netzwerkumgebung wenden Sie sich im Zweifel an Ihren Systemadministrator.

Ist das Web-IO als TCP- oder UDP-Client konfiguriert und

sucht seinerseits aktiv Verbindung zu im Netzwerk befindlichen Server-Diensten, können auch dynamische, sich ändernde IP-Adressen genutzt werden.

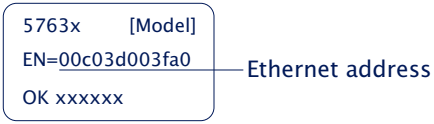
### 7.1.5 IP-Vergabe per BOOTP-Protokoll

Einige Netzwerke nutzen für die zentralisierte und dynamische Vergabe von IP-Adressen BOOTP als Vorgänger des DHCP-Protokolls. Mit den Werkseinstellungen ist BOOTP abgeschaltet. Die Aktivierung kann über das Konfigurationsmenü im Zweig *Config >> Device >> Basic Settings >> Network -> DHCP/BOOTP Enable* erfolgen. Die folgenden Parameter können übergeben werden:

- IP-Adresse
- Subnet-Mask
- Gateway-Adresse
- DNS-Server

*Zur Vermeidung ungewollter Adressvergaben oder Adressänderungen empfehlen wir, die Protokolle DHCP, BOOTP und RARP zu deaktivieren, sofern diese nicht ausdrücklich in der jeweiligen Netzwerkumgebung genutzt werden. Web-IO mit fälschlich zugeteilter IP-Adresse können nachträglich mit Hilfe der Scan-Funktion des Managementtools WuTility komfortabel gefunden und neu konfiguriert werden.*

**Adressreservierung**
Das BOOTP-Protokoll basiert auf Reservierungen von festen IP-Adressen für bestimmte Ethenet-Adressen. Das heißt, ein an das Netzwerk angeschlossenes Web-IO erhält nur dann eine IP-Adresse, wenn diese zuvor im BOOTP-Server entsprechend hinterlegt wurde. Für das Anlegen der Reservierung wenden Sie sich bitte an den zuständigen Systemadministrator. Die hierfür benötigte Ethernet-Adresse des Web-IOs kann dem am Gehäuse befindlichen Aufkleber entnommen werden.



Nachdem die notwendigen Eintragungen vom Administrator vorgenommen wurden, bezieht das Web-IO nach jedem Reset automatisch die gewünschte IP-Adresse. Um die Erreichbarkeit des Web-IO auch bei ausgefallenem BOOTP-Server zu gewährleisten, wird bei ausbleibender Antwort die bisherige IP-Adresse beibehalten.

## 7.2 Firmware Update

Die Betriebssoftware des Web-IO wird ständig weiterentwickelt. Das folgende Kapitel beschreibt aus diesem Grund das Verfahren, einen Upload der Firmware durchzuführen.

### 7.2.1 Wo ist die aktuelle Firmware erhältlich?

*Auf unserer Webseite www.WuT.de finden Sie links die„Suche über Artikel-Nr". Geben Sie hier die Artikel-Nr. Ihres Web-IO ein, wählen Sie in dem Feld darunter „Firmware" aus und klicken Sie auf „Los". Auf der Seite, die sich dann öffnet, wählen Sie den Link zur aktuellen Firmware.*

### 7.2.2 Firmware-Update über das Netzwerk mit Windows

Voraussetzung ist ein PC unter Windows 9x/NT/2000/XP mit einem Netzwerkanschluss und aktiviertem TCP/IP-Stack. Für den Update-Prozess benötigen Sie zwei Files, die, wie bereits beschrieben, auf der Homepage *http://www.wut.de* zum Download bereitstehen:

- das ausführbare *WuTility* -Tool für die Übertragung der Firmware in das Web-IO
- die Datei mit der neuen Firmware, die in das Web-IO übertragen werden soll

Das für das Update verwendete *WuTility* erkennt alle in Ihrem Netzwerk befindlichen WuT Geräte und ist weitestgehend selbsterklärend. Sollten doch Fragen oder Unklarheiten bestehen, nutzen Sie bitte die zugehörige Dokumentation oder Online-Hilfe.

*Benutzen Sie für das Update nur Wutility in der aktuellen Version . Die Verwendung von älteren Wutility-Versionen kann das Web-IO beim Update funktionell zerstören.*

*Wir empfehlen alle Passwörter zu löschen und einen Reset des Web-IO durchzuführen, bevor das Update ein-*

*gespielt wird.*

*Für das Update benötigt Wutility uneingeschränkten Netzwerkzugriff auf das Web-IO. Software-Firewalls und Internetsecuritytools sollten so konfiguriert werden, dass der Zugriff auf das Web-IO nicht behindert wird.*

*Unterbrechen Sie nie selbstständig den Update-Prozess durch Trennen der Spannungsversorgung oder Bestätigen des Reset-Tasters. Nach einem unvollständigen Update ist das Web-IO betriebsunfähig.*

Normalerweise erkennt das Web-IO selbstständig wenn die Übertragung der neuen Betriebssoftware komplett ist und führt dann selbstständig einen Reset durch. In seltenen Fällen kann es nach dem Firmware Update notwendig sein, das Web-IO per Hardware-Reset neu zu starten (Reset-Taster oder Power ON). Ein Software-Reset per Netzwerkzugriff über Port 8888 reicht dann nicht aus.

*Web-IO, die vor dem Update eine Firmware Version1.34/1.00 oder älter haben, können nur bei W&T auf den neusten Firmware-Stand gebracht werden.*

*Web-IO, die vor dem Update eine Firmware Version1.47/2.09 oder älter haben, sollten vor dem Update auf Factory Defaults zurück gesetzt werden.*

## 7.3 Notzugang

Über den seriellen Anschluss (DTE) des Gerätes steht ein Notzugang zur Verfügung. Dieser wird folgendermaßen aktiviert:

Verbinden Sie das Gerät über ein serielles Kabel mit einem PC (Null-Modem-Verbindung) und starten Sie ein serielles Terminal-Programm. Setzen Sie die Einstellungen des Programms auf

9600,8,N,1,no handshake

Drücken Sie den Reset-Taster am Gerät bzw. unterbrechen Sie kurz die Spannungsversorgung. Drücken Sie unmittelbar danach auf Ihrer Tastatur einen der folgenden Buchstaben und halten Sie die Taste solange gedrückt, bis die system error und die on error http LEDs mehrmals hintereinander aufblinken.

„u“ öffnet den Update-Port. Ein Firmware-Update kann nun eingespielt werden.

„f“ setzt das Gerät auf den Auslieferungszustand zurück. Alle vorgenommenen Konfigurations-Einstellungen (inkl. IP-Adresse) gehen verloren.

„p“ löscht alle vergebenen Passwörter.

„x“ Vergabe/Änderung der IP-Adresse. Geben Sie an der erscheinenden Eingabeaufforderung die von Ihnen gewünschte IP-Adresse ein.

## 7.4 Verwendete Ports und Netzwerksicherheit

### 7.4.1 Verwendete Ports

In der folgenden Liste finden Sie eine Übersicht, über alle im Web-IO Digital verwendeten Ports.

| Portnummer | Protokoll | Anwendung | veränderbar | abschaltbar |
|---|---|---|---|---|
| 20 | TCP | FTP-Data | 1...65535 | ja |
| 21 | TCP | FTP-Controll | 1...65535 | ja |
| 23 | TCP | Telnet [2] | nein | nein |
| 25 | TCP | SMTP Mail [1] | nein | ja |
| 37 | TCP | Anfrage an Time-Server [1] | nein | ja |
| 53 | UDP | Anfrage an DNS-Server [1] | nein | ja |
| 69 | TCP | Update über TFTP [3] | nein | ja |
| 80 | TCP | Listenport TCP/http | ja | nein |
| 110 | TCP | SMTP after POP3 [1] | nein | ja |
| 161 | UDP | SNMP [3] | nein | ja |
| 1111 | TCP | Konfiguration serieller Port [2 3] | nein | nein |
| 6000 | TCP | Telnet[2] | nein | nein |
| 7000 | TCP | FTP[2] | nein | nein |
| 8000 | TCP | Socketzugang serieller Port [2] | ja | nein |
| 8002 | TCP | Initialisierung Firmware Update [3] | nein | nein |
| 8003 | TCP | CS-Konfiguration lesen [2 3] | nein | nein |
| 8004 | TCP | CS-Konfiguraton speichern [2 3] | nein | nein |
| 8009 | TCP | Hilfsport für manuellen Update [3] | nein | ja |
| 8512 | UDP | Inventarisierung [1] | nein | nein |
| 8513 | UDP | Inventarisierung [1] | nein | nein |
| 8888 | TCP | Reset [3] | nein | nein |
| 9084 | TCP | Port-Reset [2 3] | nein | nein |
| 9094 | TCP | Box-Control-Port [2 3] | nein | nein |
| 42279 | UDP | UDP Socket | 1...65535 | ja |
| 45889 | UDP | Binary 1 | 1...65535 | ja |
| 45889 | UDP | Binary 2 | 1...65535 | ja |
| 49153 | TCP | Binary 1 | 1...65535 | ja |
| 49154 | TCP | Binary 2 | 1...65535 | ja |

[1] kein schreibender Zugriff auf das Web-IO möglich
[2] nur Web-IO 57631 mit Com-Server Funktion
[3] passwortgeschützt

*Jede Portnummer darf im Web-IO nur für einen Dienst verwendet werden. Sollen bei den veränderbaren Ports abweichende Nummern zum Einsatz kommen, ist darauf zu achten, dass keine Portnummer doppelt verwendet wird.*

### 7.4.2 Das W&T Web-IO und die Netzwerksicherheit

Die Sicherheit in Netzwerken wird heute zu Recht zunehmend

beachtet. Alle Experten sind sich darüber einig, dass es eine absolute Sicherheit beim heutigen Stand der Technik nicht geben kann. Jeder Kunde muss daher für seine konkreten Voraussetzungen ein angemessenes Verhältnis zwischen Sicherheit, Funktionsfähigkeit und Kosten festlegen.

Um hier dem Kunden eine größtmögliche Flexibilität zu ermöglichen, die sich an wechselnden Sicherheitsanforderungen von einer reinen Test- und Installationsumgebung bis hin zu kritischen Produktionsanwendungen orientiert, sind die Sicherheitsmaßnahmen in hohem Maße durch den Kunden konfigurierbar. Das vorliegende Dokument gibt einen Überblick über die Sicherheitsmaßnahmen, die auf dem Wiesemann & Theis Web-IO implementiert sind bzw. genutzt werden können. Es wird hierbei vorausgesetzt, dass die Original-Firmware von W&T (ohne kundenspezifische Anpassungen) eingesetzt wird. Weitere Details sind den jeweiligen Abschnitten der Bedienungsanleitung zu entnehmen.

### 7.4.3 Das Berechtigungskonzept des Web-IO

Wie bereits im Kapitel Web-Based Management beschrieben, kennt das Web-IO drei Stufen von Berechtigungen:

- Default User (ohne besondere Anmeldung)
- Operator
- Admin

Diese Benutzer greifen im Normalfall über die Web-Oberfläche über den bei http üblichen Port 80 zu.

Der Default User hat mit Hilfe der Web-Oberfläche lesenden Zugriff auf die Nutzdaten des Web-IO, auf die Diagnose-Tools sowie die Angaben der Online-Dokumentation (Kurz-Manual, Datenblatt des Geräts, Support-Informationen). Der Operator kann darüber hinaus die Outputs setzen und die Alarmausgaben konfigurieren. Alle anderen Einstellungen sind durch den Operator nicht veränderbar. Der User Admin hat vollen Zugriff auf alle Funktionalitäten des Geräts. Er legt die verwendete(n) Betriebsart(en) und die Netzwerkparameter fest.

Für die User *Operator* und *Admin* können Passwörter festgelegt werden. Das Gerät erzwingt weder die Verwendung von Passwörtern, noch dass diese für *Operator* und *Admin* unterschiedlich sind, noch Mindestlänge, Zusammensetzung oder regelmäßigen Wechsel der Passwörter. Dies ist bei Bedarf durch den Kunden organisatorisch sicherzustellen.

Die Übertragung des Passworts an das Web-IO geschieht ohne Verschlüsselung. Es ist also sicherzustellen, dass passwortgeschützte Zugriffe nur über ein vom Kunden als sicher betrachtetes Intranet erfolgen. Bei Zugriffen über das öffentliche Internet sind also gegebenenfalls zusätzliche Maßnahmen wie Aufbau eines VPN-Tunnels (Virtual Private Network) zu treffen. Dies ist jedoch eine generelle Problematik der Netzwerksicherheit, für die jeder Kunde entsprechende Lösungen finden muss.

### 7.4.4 Ports mit Sonderfunktion

Neben dem Zugriff über die Web-Oberfläche sind eine Reihe von Funktionen über verschiedene TCP- bzw. UDP-Ports aktivierbar. Diese sind in Tabelle 1 (Excel-Tabelle) dargestellt. Eine Reihe von Funktionen ist ein- und ausschaltbar, jedoch nur mit der Berechtigung „Admin“. Grundsätzlich ist zu empfehlen, alle nicht benötigten Funktionen abzuschalten. Näheres hierzu ist der Bedienungsanleitung zu entnehmen.

**SNMP:**
Um das Web-IO in ein SNMP-basierendes Systems Management einbinden zu können, sind alle wesentlichen Konfigurationseinstellungen auch über SNMP veränderbar. Der Zugriff ist geschützt, indem das Admin-Passwort in SNMP als *community string* verwendet wird. Der SNMP-Zugriff ist abschaltbar.

**Ports für die Inventarisierung über das Tool „WuTility“:**
Wie alle „intelligenten“ Komponenten von W&T kann das Web-IO über das Tool „WuTility“ angesprochen werden. Hierbei kann das Tool über die Ports 8512 und 8513 Informationen wie Hard- und Firmware-Version, IP-Adresse etc. auslesen.

Dieser Zugriff ist grundsätzlich nicht abschaltbar. Ein schreibender Eingriff in das Gerät ist auf diesem Wege jedoch nicht möglich.

**Firmware-Update:**
Über das Protokoll TFTP (TCP-Port 69) ist ein Firmware-Update und damit ein Überschreiben aller Konfigurations-Einstellungen möglich. Jedoch nur, wenn der Update-Modus vorher freigeschaltet wurde. Hierzu gibt es mehre Möglichkeiten:

- Einmalige Aktivierung über einen speziellen http-String in Verbindung mit dem Admin-Passwort (falls gesetzt)
- Einmalige Aktivierung, über *Config >> Session Control >> Logout* und den *Firmware Update* Button

**Reset über Port 8888:**
Durch den Aufbau einer Verbindung (z.B. mit Telnet) auf den Port 8888 kann ein Reset des Gerätes ausgelöst werden. Hierzu ist sofort nach dem Verbindungsaufbau das Admin-Passwort (falls gesetzt) anzugeben. Diese sofortige Übergabe des Passworts ist praktisch nicht manuell, sondern nur durch ein Programm möglich. Die Folgen eines Resets sind die gleichen, wie die einer kurzzeitigen Unterbrechung der Stromversorgung oder das Drücken des Reset-Tasters am Gerät.

### 7.4.5 Wiederherstellung der Factory Defaults

Durch Wiederherstellen der Werkseinstellungen (Factory Defaults) können alle vom Kunden vorgenommenen Sicherheitseinstellungen wieder rückgängig gemacht werden. Unter anderem sind dann auch die Userrechte Administrator und Operator nicht mehr durch Passwort geschützt. Diese Option stellt also zugleich einen „Notzugang“ für den Fall dar, dass das Passwort vergessen wurde oder im Notfall die Personen, die die Passwörter kennen, nicht verfügbar sind. Dieser Notzugang ist jedoch gegen unbefugten Zugriff über die Netzwerkschnittstelle geschützt:

Per Software über den Netzzugang sind die Factory Defaults nur mit Admin-Berechtigung herstellbar. Über den seriellen Zugang des Web-IO dagegen können die Factory Defaults durch eine Tastenkombination nach dem Einschalten bzw. Reset wieder hergestellt werden. Es ist also in jedem Fall darauf zu achten, dass nur berechtigte Personen physischen Zugriff auf das Gerät einschließlich der seriellen Schnittstelle erhalten, bei Bedarf also z.B. durch abschließbare Schaltschänke.

## 7.5 Timing und Grenzen bei der Signalverarbeitung

Die Systemarchitektur des Web-IO versucht bei möglichst geringem Energieverbrauch ein hohes Maß an Verarbeitungsgeschwindigkeit zu erreichen. Dabei arbeitetet das Web-IO in einem Multithreading-Verfahren, bei dem sich alle benutzten Dienste und Funktionen die Gesamtsystemleistung teilen.

### 7.5.1 Signalerkennung

Die Signalerkennung an den Inputs arbeitet mit einer besonders hohen Priorität. Es werden bis zu 500 Signalwechsel pro Sekunde sicher erkannt und gezählt. Die Signale müssen dazu min. 1ms (bei den Inputs 0 - 3 im Shortpuls-Modus 50µs) anstehen.

### 7.5.2 Signalübermittlung

Die Signalübermittlung über die Netzwerkschnittstelle arbeitet asynchron zur Signalerkennung. Das hat verschiedene Gründe: Zum einen steht je nach Netzwerkauslastung nicht immer die volle Netzwerkbandbreite zur Verfügung, zum anderen schwankt je nach Netzwerkprotokoll das Verhältnis von Protokoll-Overhead und Nutzdatenanteil (das Versenden von Signalzuständen per Email ist z.B. deutlich aufwändiger als die Übertragung über eine simple TCP-Verbindung. Abhängig von der gewählten Betriebsart nehmen die Anwendungsprozesse des Web-IO darüber hinaus mehr oder weniger Zeit in Anspruch. Das zeitliche Verhalten kann deshalb je nach Betriebsart und Protokoll erheblich variieren.

### 7.5.3 Signaltiming

Beim Timing der Signalübermittlung sind zwei Faktoren zu berücksichtigen:

- Signalabstand
  Der Signalabstand ist die Zeit, die zwischen zwei Signalwechseln mindestens nötig ist, damit beide Wechsel nicht nur erkannt (gezählt) sondern auch übertragen werden.
- Latenzzeit
  Die Latenzzeit gibt an, wieviel Zeit vom Auftreten eines Signalwechsels benötigt wird bis die entsprechende Meldung beim Kommunikationspartner ankommt.

| Zugriffs-/Betriebsart | Signalabstand Input-Erkennung | Signalabstand Output-Schalten | Typische Latenzzeit |
|---|---|---|---|
| Socketzugriff Web-IO als TCP-Server (Binary) | 50ms | 30ms | 30ms |
| Socketzugriff Web-IO als TCP-Client (Binary) | 150ms | 30ms | 30ms |
| Socketzugriff Web-IO als UDP-Peer (Binary) | 120ms | 30ms | 30ms |
| Socketzugriff Web-IO als TCP-Server (Kommandostr.) | 600ms | 30ms | 30ms |
| Socketzugriff Web-IO als TCP-Client (Kommandostr.) | 300ms | 30ms | 30ms |
| Socketzugriff Web-IO als UDP-Peer (Kommandostr.) | nur Polling | 30ms | 30ms |
| Box-To-Box | 50ms | 50ms | 30ms |
| SNMP | nur Polling | 100ms | 30ms |
| Alarm per Email | 20s | - | 20s |
| Alarm per SNMP-Trap | 10s | - | 10s |
| Alarm per Syslog | 10s | - | 10s |
| Alarm per FTP | 20s | - | 20s |
| Alarm per TCP | 2s | - | 2s |
| Alarm per UDP | 2s | - | 2s |
| Logik | 2ms | 2ms | 1ms |

*Die angegeben Zeiten sind Richtwerte für einen exklusiven Zugriff in einem lokalen, unbelasteten Netzwerk. Je nach Netzwerkumfeld und Anwendung, können die Werte erheblich abweichen - vor allem dann, wenn mehrere Betriebsarten parallel genutzt werden.*

## 7.6 Inventarisierung von W&T Geräten

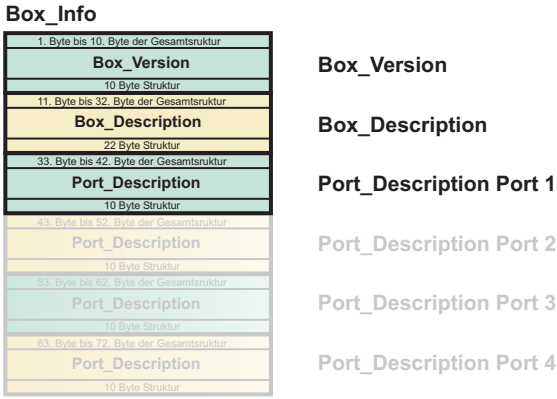
Besonders bei größeren Installationen besteht häufig die Anforderung, aktuell im Netzwerk befindliche Web-IO, aber auch W&T Com-Server zu inventarisieren. Zu diesem Zweck steht der UDP-Port 8513 zur Verfügung. Nach Empfang eines an diesen Port gerichteten Netzwerkpaketes antwortet das Web-IO dem jeweiligen Absender mit einem Infopaket. Dieses enthält Informationen über das Web-IO und dessen Netzwerk-Parameter.

Ob ein Paket per Broadcast oder direkt adressiert vom Web-IO empfangen wurde,spielt für die Generierung der Infostruktur keine Rolle. Entscheidend ist lediglich die UDP-Port Nr. 8513.

*Aus Gründen der Abwärtskompatibilität unterstützen die Web-IO auch weiterhin den bisherigen UDP-Infoport 8512. In neuen Anwendungen sollte jedoch ausschließlich der hier beschriebene Port8513 verwendet werden.*

### 7.6.1 Das Boxinfopaket

Jedes Boxinfopaket besteht aus 3-6 Datenstrukturen:



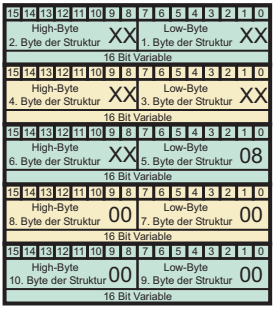

Die Einzelstrukturen Port_description für Port 2 - 4 sind optional.

Die Gesamtlänge des Infopaketes errechnet sich wie folgt: *10 + 22 + (Portanzahl × 10) Bytes.*

### Die Box-Version Struktur

*Box_Version* enthält Informationen zum jeweiligen Modell des Web-IO bzw. Com-Servers und dessen Firmwarestand.

**Box_Version**




**Version**

**SW_rev**

**HW_rev**

**Reserved1**

**Reserved2**

### Version

In Version wird die aktuell gültige Version der *Box_Info*-Struktur übergeben.

### SW_rev

Bei Web-IO Produkten: Firmware-Version der Neztwerkplatine
Bei Com-Servern: Firmware-Version des Com-Servers

### HW_rev

Bei Web-IO Produkten: Hardware-Version der Netzwerkplatine
Bei Com-Servern: Hardware-Version des Com-Servers

| HW_rev | Geräteplattform für Artikelnr.: |
|---|---|
| 0x0002 | 58001, 58004, 58201, 58301, 58304 |
| 0x0003 | 50210, 50211, 50310, 50311 |
| 0x0004 | 58211, 58411, 58412 |
| 0x0005 | 58221, 58421 |
| 0x0008 | 58631, 58431, 58620 |
| 0x0108 | 57630, 57631, 57634, 57003, 57004 |
| 0x0208 | 58031, 58634, 58231, 58331, 58334, 58633 |
| 0x0408 | 58511 |
| 0x0508 | 57007, 57008 |
| 0x0608 | 57637, 57613, 57614, 57615, 57616, 57661, 57662 |

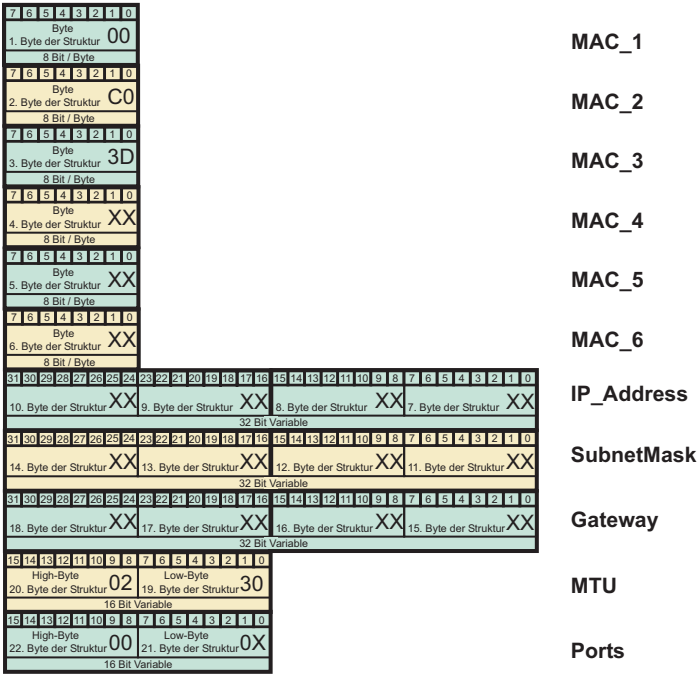
### Reserved1, Reserved2

Aktuell nicht benutzt

### Die Struktur Box_Description

Der Struktur *Box_Description* können die netzwerkrelevanten Daten wie z.B. MAC- und IP-Adresse entnommen werden.

**Box_Description**




### MAC_1 bis MAC_6

Ethernet-Adresse des Web-IO bzw. des Com-Servers

### IP_Address

IP-Adresse des Web-IO bzw. des Com-Servers

### SubnetMask

Subnet-Maske des Web-IO bzw. des Com-Servers

### Gateway

Beim Web-IO bzw. Com-Server konfiguriertes Gateway

### MTU

Maximum Transfer Unit des Web-IO bzw. Com-Servers. Netzwerkseitig größte Sendepaketgröße.

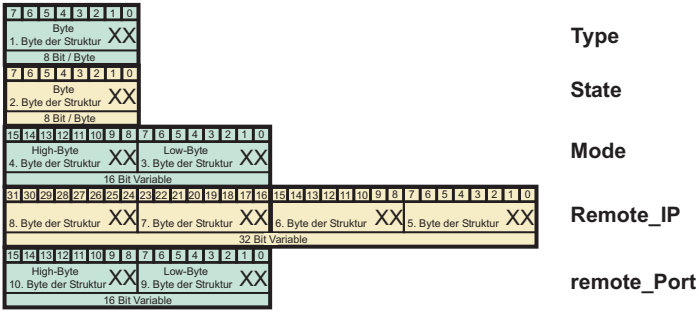
### Ports

Anzahl der verfügbaren seriellen Ports. (Beim Web-IO wird die gesamte IO-Einheit ebenfalls als ein serieller Port gezählt.)

### Port_Description

Die 3.-6. Sruktur es *Boxinfopakets* ist die Struktur *Port_Description und* liefert Informationen über die eingestellte serielle Betriebsart des Ports (nur bei W&T Com-Servern) bzw. den aktuellen Verbindungsstatus der einzelnen Schnittstellen.

**Port_Description**




Die Struktur Port_Description muss für das Web-IO nicht ausgewertet werden. Sie ist ausschließlich für W&T Com-Server betreffende Informationen gedacht.

### Type

| Type | Physikalische Porteigenschaften |
|---|---|
| 0x01 | serieller Com-Server Port (int. UART) |
| 0x02 | serieller Com-Server Port (ext. UART) |
| 0x04 | digitale E/A Com-Server Einheit |
| 0x06 | bei Com- Server Multi Port / DP RAM |
| 0x06 | Web-IO Einheit (PoE) |
| 0x07 | Web-IO Einheit |

### State

| State | Verbindungsstatus |
|---|---|
| 0x00 | Free |
| 0x01 | Connected (Verbunden) |
| 0x02 | Wait (auf Verbindung warten) |

### Mode

| Mode | serielle Com-Server Betriebsart |
|---|---|
| 0x0001 | TCP-Client |
| 0x0002 | TELNET-Client |
| 0x0003 | FTP-Client |
| 0x0004 | Box-to-Box-Client |
| 0x0005 | UDP-Peer |
| 0x0006 | Multi Port (DP-RAM, Special Protokoll) |
| 0x0007 | SNMP-Agent |
| 0x0008 | Box-toBox-Server |
| 0x0010 | SLIP-Router |
| 0x0030 | UDP Bus-Slave |
| 0x0040 | UDP-Bus-Master |

*Alle Variablen der Typen „Integer“ und „Long“ werden in Host-Order abgebildet. D.h., die niederwertigen Bytes werden als erste aufgeführt. Die IP-Adresse 172.17.2.3 erscheint z.B. in der Byte-Reihenfolge 3, 2, 17, 172.*

Die Einzel-Variablen der Struktur werden Byte-bündig übertragen. In Hochsprachen sollte das bei der Typendefinition berücksichtigt werden (C: #pragmapack(1), Delphi: packet record).

*Bei UDP-Paketen handelt es sich um verbindungslose und ungesicherte Datagramme. Besonders bei der Verwendung mit Broadcasts kann sowohl das eigene Request- wie auch das Reply-Paket des Web-IO/Com-Servers verloren gehen. Zur sicheren Ermittlung aller in einem Subnet installierten Web-IOs/Com-Server sollte der Request-Broadcast daher ggf. wiederholt werden.*

### 7.6.2 HIT - Hardware Identification Table

Anstelle der PortInfo-Struktur des BoxInfos-Paketes können bei Web-IO Produkten die Anschlusseigenschaften über das Kommando `GET /portinfo` abgerufen werden.

Der Abruf erfolgt über TCP, wobei die Kommunikation über den eingestellten HTTP-Port (ab Werk 80) abgewickelt wird.

Die Antwort des Web-IO besteht aus einem mehrzeiligen ASCII-Text. Hier beispielhaft für das Web-IO 2xDigital In, 2xDigital Out Type #57637:

```
1.0|9
22|Terminal at the bottom side||||0|0||||
2|Network||||0|0||||
3|Vdd / GND (+/-)||||0|0||||
4|Inputs 0 / GND|1|Input 0||2|2||0|10.40.22.128|00:C0:3D:BA:FF:28|80
4|Inputs 1 / GND|2|Input 1||2|2||1|10.40.22.128|00:C0:3D:BA:FF:28|80
8|Inputs 0 / GND|1|Input 0||4|4||0|10.40.22.128|00:C0:3D:BA:FF:28|80
8|Inputs 1 / GND|2|Input 1||4|4||1|10.40.22.128|00:C0:3D:BA:FF:28|80
5|Outputs 0 / GND|3|Output 0||3|3||0|10.40.22.128|00:C0:3D:BA:FF:28|80
5|Outputs 1 / GND|4|Output 1||3|3||1|10.40.22.128|00:C0:3D:BA:FF:28|80
```

Die erste Zeile enthält Informationen über die Revision der HIT-Tabelle und die Anzahl der HW-Schnittstellen. Die Syntax ist stets „X.Y|Z“.

**X**
ist die Grundversion und bestimmt den Aufbau der Struktur bzw. der Zeile, die pro Schnittstelle geliefert wird.

**Y**
Wird hochgezählt, wenn die bisher bekannte Struktur erhalten bleibt und nur neue Informationen angehängt werden.

**Z**
kündigt die Anzahl der folgenden Zeilen an. Da jede Hardwareschnittstellenbeschreibungen eine separate Zeile umfasst, ist die Gesamtzeilenanzahl Z+1.

Die folgenden Zeilen enthalten Informationen zu den verfügbaren Anschlüssen des Web-IO. Jede Zeile besteht aus 12 Informationsfeldern die durch „|“ (ASCII 124) getrennt sind.

Die Informationsfelder können Werte oder textuelle Beschreibungen enthalten. Gibt es eine entsprechende Information für den Anschluss nicht, folgt sofort das nächste Trennzeichen.

| Infofeld | Beschreibung |
|---|---|
| 1 | HIT-Nummer: Beschreibt eindeutig einen HW-Schnittstellentyp (textuelle Beschreibung der Schnittstelle in der nächsten Tabelle) |
| 2 | Werksseitige Klemmen-/Anschlussbeschreibung |
| 3 | Aktive Gruppen: Ordnet Schnittstellen durch Zuordnung gleicher Nummern in Gruppen ein, wenn sich durch Konfiguration z.B. mehrere Outputs zur Erhöhung des schaltbaren Laststroms gruppieren |
| 4 | Kundenseitig konfigurierte Klemmen-/Anschlussbeschreibung |
| 5 | Physikalische Einheit, die der Schnittstelle zugeordnet ist (z.B. °C bei Thermometern) |
| 6 | Aktuelle Software-Funktionsunterstützung (Unterstützte Kommandos usw.) |
| 7 | Abwärts kompatible Software-Funktionsunterstützung |
| 8 | Reserve |
| 9 | Nummerierung innerhalb der Gruppe (Input-Nr., Output-Nr., ....) |
| 10 | IP-Adresse über die der Anschluss / die Klemme ansprechbar ist |
| 11 | Ethernet-Adresse der Klemme |
| 12 | TCP-Port, über den der Anschluss / die Klemme ansprechbar ist |

Die folgende Tabelle zeigt die Beschreibung zu den HIT-Nummern der z.Zt. bei Web-IO eingesetzten Anschlusstypen.

| Hit-Nr. | Anschlussbeschreibung |
|---|---|
| 1 | Spannungsversorgung: DC 18 ... 48V (+/-10%), AC 18 ... 30Veff (+/-10%), keine besondere Polung erforderlich |
| 2 | Netzwerk: RJ45, 10/100BaseT, Autosensing, POE |
| 3 | Versorgungsspannung für digitale Outputs: DC 6-30V |
| 4 | Digitaler Eingang: Spannungsbereich -30 ... +30V DC, Schaltschwelle 8V +/-1V, "Ein"-Strom 2.2mA |
| 5 | Digitaler Ausgang: DC 6-30V, 500mA |
| 6 | Spannungsversorgung: DC 24 ... 48V (+/-10%), AC 18 ... 30Veff (+/-10%), DC-Polung beachten |
| 7 | RS232-Schnittstelle als Konfigurationszugang (9600,8,N,1) |
| 8 | Zähler: Spannungsbereich -30 ... +30V DC, Schaltschwelle 8V +/-1V, "Ein"-Strom 2.2mA, max. Frequenz 1kHz, Auflösung 31 Bit |
| 9 | Temperatursensor, PT1000, Messbereich -40 ... +85°C |
| 10 | Luftfeuchtesensor, Messbereich 0 ... 100% rF |
| 11 | Luftdrucksensor, Messbereich 10 ... 1100hPa |
| 12 | Temperatursensor, NTC (10k), Messbereich -45 ... +75°C |
| 13 | Temperatursensor, PT100/1000, Messbereich -200 ... +650°C |
| 14 | Spannungsversorgung: AC/DC 12 ... 24V, keine besondere DC-Polung erforderlich |
| 15 | Netzwerk: RJ45, 10/100BaseT, Autosensing |
| 16 | Digitaler Ausgang: AC/DC 39V, AVG 300mA (peak 500mA), AC/DC 11.7W |
| 17 | Spannungsversorgung: DC 12 ... 48V (+/-10%), AC 12 ... 30Veff (+/-10%), keine besondere Polung erforderlich |
| 18 | Stromeingang: 0 ... 20mA, Auflösung: 2,5uA |
| 19 | Spannungseingang: 0 ... 10V, Auflösung: 1,25mV |
| 20 | Stromausgang: 0 ... 20mA |
| 21 | Spannungsausgang: 0 ... 10V |
| 22 | Spannungsversorgung: DC 24 ... 48V (+/-10%), AC 18 ... 30Veff (+/-10%), keine besondere Polung erforderlich |
| 23 | RS232-Schnittstelle: max. 57,6kBaud |
| 24 | Digitaler Ausgang: Relais, 30V/5A DC, 48V/5A AC |

In Infofeld 6 ist festgelegt, welche Software-Funktionen unterstützt werden.

| Nr | Typ | verfügbare Funktionen |
|---|---|---|
| 1 | analoger Eingang | Lesen eines einzelnen analogen Eingangs<br>(GET /single**x**)<br>Lesen aller analogen Eingangswerte des Gerätes<br>(GET /single) |
| 2 | digitaler Eingang | Lesen eines einzelnen digitalen Eingangs<br>(GET /input**x**?PW=**passwd**&)<br>Lesen des kompletten Eingangs-Bitmusters des Gerätes<br>(GET /input?PW=**passwd**&) |
| 3 | digitaler Ausgang mit Lese- u. Schreibzugriff | Lesen eines einzelnen digitalen Ausgangs<br>(GET /output**x**?PW=**passwd**&)<br>Lesen des kompletten Ausgangs-Bitmusters des Gerätes<br>(GET /output?PW=**passwd**&)<br>Setzen eines einzelnen digitalen Ausgangs<br>(GET /outputaccess**x**?PW=**passwd**&State=**ON**/**OFF**&) |
| 4 | Zähler | Lesen eines einzelnen Zählerstandes<br>(GET /counter**x**?PW=**passwd**&)<br>Lesen aller Zählerstände des Gerätes<br>(GET /counter?PW=**passwd**&)<br>setzen eines einzelnen Zählerstandes<br>(GET /counterclear**x**?PW=**passwd**&Set=**value**&) |
| 5 | digitaler Ausgang, nur Lesezugriff | Lesen eines einzelnen digitalen Ausgangs<br>(GET /output**x**?PW=**passwd**&)<br>Lesen des kompletten Ausgangs-Bitmusters des Gerätes<br>(GET /output?PW=**passwd**&) |
| 6 | analoger Ausgang | Lesen eines einzelnen analogen Ausgangs<br>(GET /single**x**)<br>Lesen aller analogen Ausgangswerte des Gerätes<br>(GET /single)<br>Setzen eines einzelnen digitalen Ausgangs<br>(GET /outputaccess**x**?PW=**passwd**&State=**y.z**&) |

*Wenn für einen physikalischen Anschluss mehr l eine Funktion unterstützt wird (z.B. Input-Status und Counter), wird er in der HIT-Tabelle für jede Funktion separat - also ggf. mehrfach - aufgeführt.*

Das Kommando *GET /portinfo* wird bei Web-IO Digital Geräten ab Firmware 2.26 bzw 3.26 unterstützt.

## 7.7 Fehlermeldungen

Die hier gezeigte Liste gibt einen Überblick über alle Fehlermeldungen, die im Report unter *Diag >> Report* angezeigt werden können

| Fehler-Nr. | Fehlermeldung |
|---|---|
| 0 | Gerätestatus: OK |
| 1 | System: Netzwerkstörung oder Fehler in der Konfiguration. |
| 2 | DNS: Formatfehler in der DNS Anfrage. |
| 3 | DNS: Interner DNS Server Fehler. |
| 4 | DNS: Name existiert nicht. |
| 5 | DNS: DNS-Anfragetyp wird nicht unterstützt. |
| 6 | DNS: DNS-Anfrage wurde abgewiesen. |
| 7 | DNS: DNS Server ist nicht erreichbar. |
| 8 | DNS: DNS Server Antwort nicht erkannt. |
| 9 | DNS: Falsche IP-Adresse. |
| 10 | DNS: keine Auflösung möglich. |
| 11 | Mail: Server verweigert Zugriff. |
| 12 | Mail: Übermittlung des Absenders fehlgeschlagen. |
| 13 | Mail: Übermittlung der Empfänger fehlgeschlagen. |
| 14 | Mail: Wechsel in den Datenmodus fehlgeschlagen. |
| 15 | Mail: Übermittlung des Mailtextes fehlgeschlagen. |
| 16 | Mail: Schließen der Mailserververbindung fehlgeschlagen. |
| 17 | Mail: Mail Server antwortet nicht. |
| 18 | Mail: IP-Adresse oder Name des Mailservers nicht eingegeben. |
| 19 | Mail: Das Versenden der Mail wurde abgebrochen und wird wiederholt. |
| 20 | Übertragungsfehler: Fehlercode #0. |
| 21 | Protokollfehler: Fehlercode #0. |
| 22 | System: Es wurde kein Router konfiguriert. |
| 23 | System: Ziel IP-Adresse unbekannt: 0.0.0.0. |
| 24 | System: Es wurde eine Netzwerkstörung erkannt (Kabel offen o. kein Link). |
| 25 | System: Der COM-Server hat einen Reset ausgelöst. |
| 26 | Warnung: Watchdog Timer Reset ausgelöst. |
| 27 | UDP: UDP Portnummer ungültig: verwende Port 42279. |
| 28 | System: Checksummenfehler: Factory Defaults wurden geladen. |
| 29 | Snmp: IP-Adresse oder Name des Snmp Managers nicht eingegeben. |
| 30 | UDP Alarm: IP-Adresse oder Name des UDP Servers nicht eingegeben. |
| 31 | UDP Alarm: Port des UDP Servers nicht eingegeben. |
| 32 | System: Anzahl der Pulse übersteigt 2000/sek: Pulsmode Input 0 gestoppt! |
| 33 | System: Anzahl der Pulse übersteigt 2000/sek: Pulsmode Input 1 gestoppt! |
| 34 | System: Anzahl der Pulse übersteigt 2000/sek: Pulsmode Input 2 gestoppt! |
| 35 | System: Anzahl der Pulse übersteigt 2000/sek: Pulsmode Input 3 gestoppt! |
| 36 | Binary 2: Kein oder ungültiger Destination Port. |
| 37 | Binary 2: Keine oder ungültige Destination IP-Adresse. |
| 38 | System: Watchdog Timer abgelaufen! |
| 39 | Binary 1: Kein oder ungültiger Destination Port. |
| 40 | Binary 1: Keine oder ungültige Destination IP-Adresse. |
| 41 | System: Portnummer mehrfach vergeben! |
| 42 | Binary 2: ungültige lokale Portnummer! |
| 43 | Binary 1: ungültige lokale Portnummer! |
| 44 | Box2Box: Zugriff vom Slave abgewiesen, bitte Box2Box-Einstellungen kontrollieren |
| 45 | Box2Box: Slave befindet sich im Box2Box-Mastermode. |
| 46 | Box2Box: Master konnte nicht gelöscht werden, bitte ausloggen. |
| 47 | System: Lastabschaltung: Details unter Diag>>Test>OutputConfig. |
| 48 | System: Neustart durch Report löschen. |
| 49 | Reservierte Portnummer: bitte den Anhang der Anleitung beachten. |
| 50 | Time Server: Synchronisation fehlgeschlagen. |
| 51 | System: Data Overrun Error! Daten im Eingangspuffer wurden überschrieben. |

| Fehler-Nr. | Fehlermeldung |
|---|---|
| 52 | TCP Client Alarm: IP-Adresse oder Name des TCP Servers nicht eingegeben. |
| 53 | TCP Client Alarm: Port des TCP Servers nicht eingegeben. |
| 54 | TCP Client Alarm: Server nicht erreicht. |
| 55 | Syslog Alarm: IP-Adresse oder Name des Syslog Servers nicht eingegeben. |
| 56 | Syslog Alarm: Port des Syslog Servers nicht eingegeben. |
| 57 | Syslog Alarm: Server nicht erreicht. |
| 58 | Reservierte Portnummer: Einstellung auf Port 80 geändert. |
| 59 | Mail: IP-Adresse oder Name des POP3-Servers nicht eingegeben. |
| 60 | FTP Alarm: IP-Adresse oder Name des FTP Servers nicht eingegben. |
| 61 | FTP Alarm: Control Port des FTP Servers nicht eingegeben. |
| 62 | FTP Alarm: Data Port des FTP Servers nicht eingegeben. |
| 63 | FTP Alarm: Fehler beim Datentransfer. |
| 64 | FTP Alarm: keinen User-Eintrag gefunden. |
| 65 | FTP Alarm: kein Passwort eingegeben. |
| 66 | FTP Alarm: keinen Account-Eintrag gefunden. |
| 67 | FTP Alarm: Kein Filename eingetragen. |
| 68 | FTP Alarm: Server antwortet nicht. |
| 69 - 78 | Fehlernummern in Web-IO Digital nicht unterstützt |
| 79 | HTTP Client: IP-Adresse oder Name des HTTP Servers nicht eingegeben. |
| 80 | HTTP Client: Port des HTTP Servers nicht eingegeben. |
| 81 | Zählpulse konnten nicht einer Richtung zugeordnet werden. |
| 82 | System: Speicherung der Messwerte im Flash fehlgeschlagen |
| 83 | Uhrenmodul: Batterie ist leer. |
| 84 | Speicherüberlauf: Neue Einstellungen konnten nicht übernommen werden. |
| 85 | Motherbox Client: IP-Adresse oder Name des Motherbox Servers nicht eingegeben. |
| 86 | Motherbox Client: Port des Motherbox Servers nicht eingegeben. |

### 7.8.1 Technische Daten 57630 / 57631

| | |
|---|---|
| Netzwerk: | Ethernet<br>10/100BaseT autosensing |
| Protokoll: | TCP und UDP Sockets, Client und Server<br>SNMP inkl. Traps<br>SMTP (E-Mail), FTP, Syslog<br>OPC-Server<br>Inventarisierung, Gruppenmanagement |
| Latenzzeiten: | Das Web-IO 12xDigital ist für Signalwechselzeiten > 50ms konzipiert. Das Zeitliche Verhalten steht darüber hinaus sehr stark in Abhängigkeit mit der Netzwerkauslastung und der Netzwerkinfrastruktur. |
| Digitale Ausgänge: | Gruppe 12 x Digital Out 6V-30V, 500mA<br>Gruppenbildung à 2 oder 4 Ausgänge<br>max. Gruppenstrom 2A<br>max. Gesamtstrom 6A<br>Mindestlast 22KOHM<br>Kurzschlussfest durch thermische Sicherung |
| Digitale Eingänge: | 12 x Digital In,<br>max. Eingangsspannung +/-30V<br>verpolungssicher innerhalb dieses Bereichs<br>Schaltschwelle 8V +/- 1V<br>"Ein"-Strom = 2,2 mA |
| Anschlüsse: | 2 x 16-fach Schraubklemmen |
| Galvanische Trennung: | Digital-Ausgänge - Netzwerk: min. 500 V zum Netzwerk<br>Digital-Eingänge: min 1000 V |
| Serieller Port: | Konfigurationsport mit RS232-Schnittstelle<br>DB9-Stecker, Belegung wie PC-COM |
| Baudrate: | 9600 Baud |
| Datenformat: | 8 Datenbit, 2 Stopbit<br>No Parity |
| Flusskontrolle: | No Handshake |
| Anzeigen: | Status-LEDs Netzwerk<br>24 LEDs für digitale Zustände |
| Bedienungselemente: | 1 verdeckt angebrachter Reset-Taster |
| Stromversorgung: | 12-24V AC/DC<br>Stromaufnahme bei 12V ca. 200mA<br>Stromaufnahme bei 24V ca. 100mA |
| Lagertemperatur: | -25°C - 70°C |
| Betriebstemperatur: | 0°C - 60°C |
| Gehäuse: | Kunststoff-Gehäuse zur Hutschienen-Montage<br>106,8 x 87,8 x 62,6 (l x b x h) |
| Gewicht: | ca. 260 g |

### 7.8.2 Technische Daten 57637, 57638

| | |
|---|---|
| Netzwerk: | Ethernet<br>10/100BaseT autosensing |
| Protokoll: | TCP und UDP Sockets, Client und Server<br>SNMP inkl. Traps<br>SMTP (E-Mail), FTP, Syslog<br>OPC-Server<br>Inventarisierung, Gruppenmanagement |
| Latenzzeiten: | Das Web-IO 2xDigital ist für Signalwechselzeiten > 50ms konzipiert. Das Zeitliche Verhalten steht darüber hinaus sehr stark in Abhängigkeit mit der Netzwerkauslastung und der Netzwerkinfrastruktur. |
| Digitale Ausgänge: | 2 x Digital Out 6V-30V, 500mA<br>Gruppenbildung à 2 zur Erhöhung des max. Schaltstromes auf 1A<br>Mindestlast 22KOHM<br>Kurzschlussfest durch thermische Sicherung |
| Digitale Eingänge: | 2 x Digital In,<br>max. Eingangsspannung +/-30V<br>verpolungssicher innerhalb dieses Bereichs<br>Schaltschwelle 8V +/- 1V<br>"Ein"-Strom = 2,2 mA |
| Anschlüsse: | 6-fach Schraubklemmen |
| Galvanische Trennung: | Digital-IOs - Netzwerk: min. 1000 V zum Netzwerk |
| Anzeigen: | Status-LEDs Netzwerk |
| Bedienungselemente: | keine |
| Stromversorgung: | PoE - Power over Ethernet oder<br>24 - 48V AC/DC auf separater 2fach Schraubklemme<br>Stromaufnahme bei 12V ca. 200mA<br>Stromaufnahme bei 24V ca. 100mA |
| Lagertemperatur: | -25°C - 70°C |
| Betriebstemperatur: | 0°C - 60°C |
| Gehäuse: | Kunststoff-Gehäuse zur Hutschienen-Montage<br>105 x 75 x 22 mm (l x b x h) |
| Gewicht: | ca. 125 g |

### 7.8.4 Technische Daten 57634

| | |
|---|---|
| Netzwerk: | Ethernet<br>10/100BaseT autosensing |
| Protokoll: | TCP und UDP Sockets, Client und Server<br>SNMP inkl. Traps<br>SMTP (E-Mail), FTP, Syslog<br>OPC-Server<br>Inventarisierung, Gruppenmanagement |
| Latenzzeiten: | Das Web-IO 12xDigital ist für Signalwechselzeiten > 50ms konzipiert. Das Zeitliche Verhalten steht darüber hinaus sehr stark in Abhängigkeit mit der Netzwerkauslastung und der Netzwerkinfrastruktur. |
| Digitale Ausgänge: | 6 Potentialfeie Kontakte (Schließer)<br>Schaltspannung max 48V AC bei 5A Laststrom<br>Schaltspannung max 30V DC bei 5A Laststrom<br>Pause zwischen zwei Schaltvorgängen min. 200ms |
| Digitale Eingänge: | 12 x Digital In,<br>max. Eingangsspannung +/-30V<br>verpolungssicher innerhalb dieses Bereichs<br>Schaltschwelle 8V +/- 1V<br>"Ein"-Strom = 2,2 mA |
| Anschlüsse: | 2 x 16-fach Schraubklemmen |
| Galvanische Trennung: | Digital-Ausgänge - Netzwerk: min. 500 V zum Netzwerk<br>Digital-Eingänge: min 1000 V |
| Serieller Port: | Konfigurationsport mit RS232-Schnittstelle<br>DB9-Stecker, Belegung wie PC-COM |
| Baudrate: | 9600 Baud |
| Datenformat: | 8 Datenbit, 2 Stopbit<br>No Parity |
| Flusskontrolle: | No Handshake |
| Anzeigen: | Status-LEDs Netzwerk<br>24 LEDs für digitale Zustände |
| Bedienungselemente: | 1 verdeckt angebrachter Reset-Taster |
| Stromversorgung: | 12-24V AC/DC<br>Stromaufnahme bei 12V ca. 300mA<br>Stromaufnahme bei 24V ca. 150mA |
| Lagertemperatur: | -25°C - 70°C |
| Betriebstemperatur: | 0°C - 60°C |
| Gehäuse: | Kunststoff-Gehäuse zur Hutschienen-Montage<br>106,8 x 87,8 x 62,6 (l x b x h) |
| Gewicht: | ca. 260 g |

## 7.9 Entsorgung

Web-IO Geräte dürfen nicht über den normalen Hausmüll entsorgt werden, sondern sind einer offiziellen Sammelstelle für Elektroschrott zuzuführen.

Web-IO des Types 57634,57637 und 57638 enthalten eine nicht aufladbare Lithium-Knopfzelle Typ BR (Lithium-Kohlenstoffmonofluorid) zum Erhalt der Uhrzeit auch bei ausgeschaltetem Gerät, welche nach Ablauf der Lebensdauer gesondert entsorgt werden muss.

Bringen Sie diese zum Recycling zu einer offiziellen Sammelstelle für Batterien.

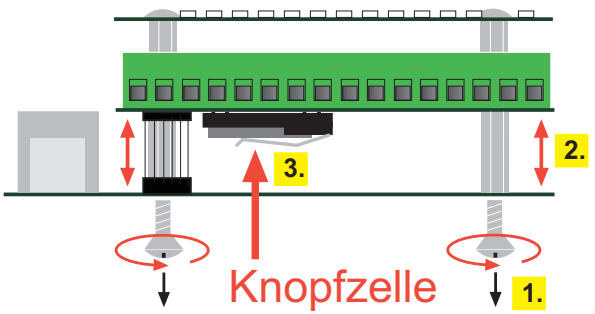
Zur Entnahme der Batterie trennen Sie zunächst das Gerät vn allen angeschlossenen Kabeln und Sensoren.

Lösen Sie die vier Klammern an der Gehäuseseite und öffnen Sie das Gehäuse.

Entfernen Sie die beiden Schrauben mit denen die beiden unteren Platinen miteinander verbunden sind und entfernen Sie die untere Platine.

Auf der Rückseite der mittleren Platine befindet sich die Knopfzelle. Entfernen Sie diese aus der Halterung und führen Sie sie dem Recycling zu.

Seitenansicht



# Index

## Symbole


# 280
10/100BaseT 39
127.0.0.1 155
50xxx-Mode 214


## A


Administrator 47
Adressreservierung 288
AJAX 173
Alarm bei Kaltstart 145
Alarm bei Warmstart 146
Alarm über UDP 152
Alarmbedingungen 140
Alarmmeldung 156
Alarmzustand 141
AllCounter 234
Anschlussbeispiel 24
Anschlüsse 22
Application Mode
72, 76, 115, 118, 216, 222
ARP-Kommandos 280
ASCII 189
Ausgangsspannungsversorgung 27
Authentification 148
Auto Negotiation 39


## B


Batterien 313
Beenden des Box-to-Box Modus 78
Berechtigungskonzept 294
Beschreibung 65
Beschriftungen 65
Binärstrukturen 189, 190, 213
BINARY 189, 213
BinInfo 223
BOOTP 53
BOOTP-Protokoll 288
BOOTP-Protoll 280
Box-to-Box 10, 70
Browser 167, 172, 177
Browser Zugriff 10


## C


C 189
C++ 189
Change Detection 256
ClearCounter 235
clearCounter 181
ClearDiagnosis 237
Client HTTP Port 72, 114, 117, 216
CommandLoop 178
Community String 100
Compatible 50xxx 220
Config.xml 267
Contact 56
Counter 233
counterpolling 184


## D


DataRequest 179
Date 246
Datum 246
Default User 47
Delphi 189
device 183
Device Name 55
Device Text 56
Device Type 265
DHCP 18, 53
DHCP-Protokoll 284
Diag Messages 100
Diagnose 271
Diagnosis 236
DNS Server 53
Download 267
DstPort 225
Dummy 224
Duration 263

Dynamische IP-Adressen 287
Dynamische Webseiten 173

**E**
E-Mail 146
E-Mail Addr. 149
Eigene Webseiten 10
Einzelereignis 141
Email 12, 140
Entsorgung 313
ERP 239
ERP-Gateway 239
Error-LED 42
Ethernet-Anschluss 39
Exception Code 132

**F**
Factory Defaults 296
Fehler-Report 271
Fehlermanagement 209
Fehlermeldungen 209, 308
File Name 163
Firmware Update 290
FTP 162
FTP Alarm Text 163
FTP Control Port 165, 242
FTP Local Data Port 163
FTP Server IP 165
FTP-Account 165
FTP-Server 162
Function Code 120, 121

**G**
Galvanische Trennung 39
Gateway 52, 282
Gerätenamen 65
Gerätetyp 111
Google-Kalender 87
Google-Konto 85
Group Mode 258, 274
Grundeinstellung 51

**H**
Header 197
HextoInt 178
Hilfspannung 25
Hochsprachen 189
Home-Seite 60
home.htm 167, 268
HTML 175
HTML Upload 269
HTTP 60
HTTP Client 202
HTTP-Port 68
HTTP-Requests 12, 175
HTTPLogin[n] 224

**I**
iCal 90
iCal Datei 268
iCal-Datei 85
Inactive Timeout 76, 219
Inactivity Timeout 203, 257
Input Inverter 262
Input Mask 261
Input Mode 256
Input Trigger 72, 76, 114, 117, 197, 199, 203, 216, 219, 222
Input-Beschaltung 23, 30, 35
Input-Konfiguration 255
Inputmustern 198
inputpolling 183
Integration 99
Interval 76, 200, 219, 222
Inventarisierung 300
IO-Strukturen 226
IP Addr 52, 154, 160
IP-Adresse 18, 51
IP-Adressvergabe 280

**J**
Java-Applets 182
JavaScript 173, 185

## K
Kalender 94
Keep Alive Timeout 54
Klemmen 24
Kommandosstrings 191
Kommandostrings 189

## L
Lease-Time 286
LED -Test 276
Length 120
length 227
Link-Status 40
Links und Rücklinks 172
Load Contro 275
Local Port
72, 75, 114, 117, 203, 218, 221
localhost 155
Location 56
Logic Func 262
Logik Funktionen 260
Logo 67, 266
logo Upload 269

## M
Mail Server 148
Mailtext 149
Manager IP 100, 151
Modbus 99
Modbus Speicheraufteilung 138
Modbus TCP 11, 116
Modbus TCP Kommunikation 119
Modbus-Adressbereiche 133
Modbus-Alarm-Auslösung 135
Modbus-Zugriff 63
Mode 219, 248
Month 248

## N
Navigation 46
Netzteile 22
Netzwerkanschluss 16
Netzwerkbuchse 39
Netzwerkparameter 45
Netzwerksicherheit 294
Notzugang 292

## O
on error 43
Online-Kalender 85
OPC 11, 99
OPC - Standarisierter Zugriff 109
OPC Device 115
OPC-Device 112
OPC-Gerätename 111
Operator 48
Option 164
Options 237
Output 27
Output Inverter 262
Output verändern 104
Output-Beschaltung 26, 32, 37
Output-Konfiguration 273
outputaccess 155
outputpolling 184

## P
Password 148, 165
Passwort-Schutz 223
Passwörter 49
Passwortschutz 185
Pinbelegung 41
PoE 16, 22, 39
Polarität 29
pollen 111
pollingrate 185
POP3 Server 148

Port 154, 158, 160, 206
Port Bezeichnung 66
Power-LED 42
PPS 239
Private-MIB 99
Profile 47
Protocol Identifier 120
Puls Polarity 257, 263
Pulsmode 262

## R

RARP-Protokoll 280
RARP-Server 283
Read Coils 121
Read Discret Inputs 121
Read Discrete Inputs 123
Read Excaption State 121
Read Exception State 131
Read Holding Registers 121, 123
Read Input Register 121
Read Input Registers 125
Read Register 228
ReadAllCounter 233
ReadClearAllCounter 234
ReadClearCounter 232
ReadCounter 232
ReadDiagnosis 235
Refresh 90
RegisterRequest 230
RegisterState 230
Remote IP Addr 221
Remote Port 206, 221
Reply Address 147
Reset 296

## S

Safety State 253, 258
Safety Timeout 252
Schaltuhr 84
Schaltvorgaben 85
Schaltzeiten 85, 91
SendMode 231
serielle Schnittstelle 280
Serieller Anschluss 41
Server HTTP Port 76, 218
Server IP Addr 76, 218
Server IP-Adress 203
Server Password 76, 219
Server Port 203, 218
Server URL 203
ServerPort 75
Session Login 102
SetBit 229
setOutput 181
setPassword 182
Short Puls Detection 256
showerrors 183
Slave 77
Slave Web-IO 70
Smart-Seite 64
smart.htm 167, 268
Smartphone 10
SNMP 11, 99
SNMP Alarm 150
SNMP-Zugriff 99
Socket Device 220
Socket-Client 217
Socket-Server 215
Socketprogramme 213
Sommerzeit 247
Sonderfunktion 295
Spannungsversorgung 22
SrcPort 225
Start Address 121
Start_1 227
Start_2 227
StructLength 227
strukt_typ 227
Subject 149
Subnet Mask 52
Subnet-Mask 282
SubType 225

Sunbird 94
Sync. Time 249
Syslog 159
Syslog-Server 159
Syslog-Text 161
system error 43
System Name 286
System Traps 100

## T

TCP Kommunikation 200
TCP-Text 158
TCP-Verbindung 194
Technische Daten 310
Termineinzelheiten 95
Time 246, 248
Time Swich 262
Time Switch 84
Time Switch File Upload 270
Time Trigger 203
Time-Server 248
Transaction Identifier 120
Trap-Text 151
Trigger 141
Type 224

## U

Überlast 143
UDP-Peer 221
UDP-Text 154
Uhrzeit 246
Unit identifier 120
updateDisplay 180
Upload 267
User 165
User-Seite 61
user.htm 167, 268
UTC 247
UTC-Server 249
UTCOffset 247

## V

Vergabe der IP-Adresse 18
Versorgung 23
Versorgungsspannung 16
Versorgungsspannungseingang 29
Verwendete Ports 293
Virtuelle Register 134
Visual Basic 189
Visual Basic for Applications 189
Vorschau 97

## W

W&T Tags 168
Wayback 239
Wayback Response 242
Wayback-Zugriff 241
Web Based Management 285
Web-Based Management 45
Weekday 248
Winterzeit 247
Write Multiple Coils 121, 127
Write Multiple Register 121
Write Multiple Registers 129
Write Single Coil 121, 126
Write Single Register 121
WriteRegister 228
WuTility 285
Wutility 18

## Z

Zeitgesteuerter Alarm 144
Zeitgesteuertes Schalten 10
Zeitzonen 246
Zugriffspasswörter 48
Zugriffsrechte 47
Zuleitungen 31