Thema:

Web-IO - eigene Anwendung im Browser

IO-Signale mit HTTP-Requests abfragen und setzen

Der große Vorteil von browserbasierten Anwendungen ist die Unabhängigkeit vom Betriebssystem des Anwendungsrechners. Egal ob Windows, Linux, iOS oder Android - der Browser gehört zum System. Selbst rein technische Embedded Systems ohne grafische Benutzeroberfläche bieten heute Mechanismen, um HTTP-Requests zu versenden und die entsprechenden Antworten zu empfangen.

Auf dieser Seite wird zunächst das prinzipielle Ansprechen unserer Web-IO-Produkte über HTTP-Requests bzw. HTTPS-Requests gezeigt, unterteilt in die Bereiche:

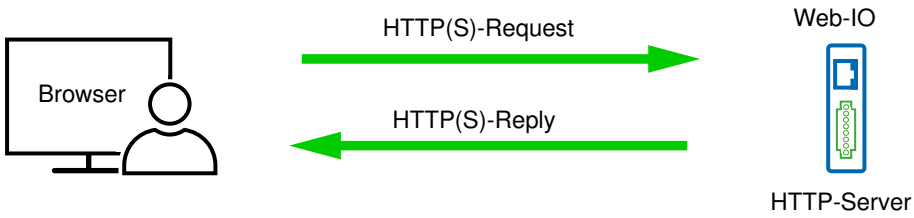
- Zugriff über HTTP-Requests
- Web-API konfigurieren
- Aufbau und Struktur der HTTP-Requests

Zugriff

Zugriff über HTTP-Requests

An dieser Stelle wollen wir nur das beschreiben, was für den Umgang mit HTTP-Requests grundsätzlich erforderlich ist. Eine weiterführende Beschreibung von HTTP bzw. HTTPS finden Sie im Buch TCP/IP-Ethernet, das Sie hier kostenlos herunterladen können.

Wer Anwendungen erstellt, die HTTP nutzen, muss sich um die Abwicklung auf Protokollebene meist nicht selbst kümmern. Bei Anwendungen im Browser übernimmt das der Browser selbst oder zugelinkte Funktionsbibliotheken. Auch moderne Hochsprachen bieten Funktionsaufrufe, denen nur noch der entsprechende HTTP-Request übergeben wird.



Durch den Aufruf von https://wut.de/webio in der Adresszeile des Browsers liefert der Webserver von W&T zum Beispiel eine Webseite mit der Übersicht verfügbarer 230V Web-IO-Modelle zurück.

Ein einfaches Beispiel für einen technischen HTTP-Request ist der Aufruf von https://klima.wut.de/single, der einfach nur die aktuelle Außentemperatur am W&T-Firmengebäude zurückgibt.

Aufbau eines HTTP-Requests

Ein HTTP-Request beginnt immer mit dem Schlüsselwort http:// bzw. https://, wenn verschlüsselte Übertragung gefordert ist. Dahinter folgt die IP-Adresse oder der Hostname des Webservers, der angesprochen werden soll.

HTTP wird über TCP-Port 80 bzw. HTTPS über TCP-Port 443 abgewickelt. Wenn der Server unter einem abweichenden Port erreichbar ist, kann dieser mit : getrennt angehangen werden. Erst jetzt folgt getrennt mit/ der eigentliche Request (z.B. der Name einer Webseite wie index.html). Ggf. notwendige Parameter können hinter einem ? folgen. Bei mehr als einem Parameter wird als Trenner & verwendet.

```
http[s]://<server-ip>[:<port>]/<request>[?parameter&parameter&.....]
```

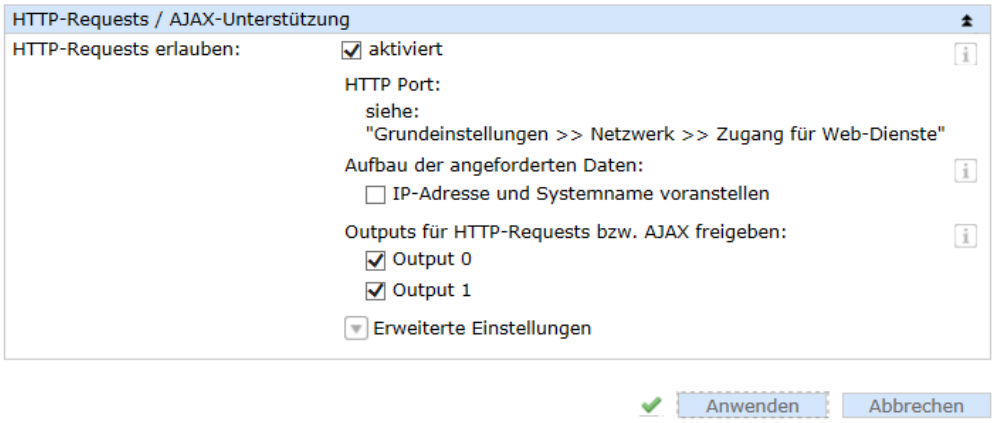
Kommunikation über HTTP-Requests

Datenaustausch über HTTP-Requests gleicht immer einem Frage-/Antwort-Szenario. Ein HTTP-Client schickt einen Request zu einem Webserver und der Webserver antwortet mit einem HTTP-Reply. Aktiver Teil einer HTTP-Kommunikation ist also immer der HTTP-Client. Der Webserver kann über reine HTTP-Lösungen aus eigener Initiative niemals Daten zum Client senden.

Konfiguration

Web-API konfigurieren

Wählen Sie im Menübaum der Web-Oberfläche *Kommunikationswege >> Web-API* und aktivieren Sie *HTTP-Requests erlauben*.



Der *HTTP-Port* kann bei Bedarf unter *Grundeinstellungen >> Netzwerk* angepasst werden.

Die Outputs, die über die Web-API geschaltet werden sollen, müssen entsprechend freigegeben werden.

Auf den Punkt *IP-Adresse und Systemnamen voranstellen* wird später noch eingegangen.

Aufbau der Requests

Aufbau und Struktur der HTTP-Requests

Wie bereits weiter oben gezeigt, beginnt jeder HTTP-Request mit `http://` bzw. `https://` gefolgt von dem eigentlichen Request-Kommando. Die hier gezeigte Groß-/Kleinschreibung ist genau so umzusetzen. Mit `?` abgetrennt können Parameter folgen. Beim Web-IO Digital auf jeden Fall immer das Passwort mit der Syntax `PW=`*password*`&`. Jeder Parametersatz wird mit `&` abgeschlossen, so dass zwei Parametersätze auch durch `&` getrennt werden.

Beispiel: Input-Abfrage

Als Beispiel hier das Kommando zum Abruf der Input-Schaltzustände. Das Web-IO ist mit dem Passwort*blau* geschützt.

`http://192.168.0.25/input?PW=blau&`

Die Antwort des Web-IO sieht je nach Modell zum Beispiel so aus:

`input;1` (Web-IO Digital 2xIn, 2xOut)
`input;0001` (Web-IO Digital 12xIn, 12xOut bzw 12xIn, 6xRelais)

Der Aufbau besteht aus dem Schlüsselwort der Anfrage, mit Semikolon getrennt gefolgt von einem oder mehreren Werten.

Hier beginnt die Antwort also mit dem Schlüsselwort `input` und es folgt der Wert `1` bzw. `0001`, der in hexadezimaler Schreibweise den Schaltzustand aller Inputs wiedergibt. Hier konkret Input 0 = ON, alle anderen Inputs = OFF.

Zum besseren Verständnis der hexadezimalen Input-Codierung hier nochmal ein Beispiel für ein Web-IO mit 12 Inputs:

Im Zustand ON sind die Inputs 0,1,5,7,10,11. Die anderen Inputs sind im Zustand OFF. Aus diesem Bit-Muster ergibt sich eine Dualzahl, bei der ON-Bits = 1 sind und OFF-Bits = 0.

Hier also `1100 1010 0011`.
In hexadezimaler Schreibweise entspricht das `CA3` bzw. in vierstelliger Darstellung `0CA3`.
Mehr zu hexadezimalen Zahlen finden Sie hier.

Beispiel: Setzen der Outputs

Das Kommando zum Setzen von Outputs hat einen ähnlichen Aufbau wie das `input`-Kommando, hat aber mehr Parameter:

`http://192.168.0.25/outputaccess0?PW=blau&State=ON&`

Die Ziffer hinter dem Schlüsselwort `outputaccess` gibt an, welcher Output geschaltet werden soll - hier Output 0. Mit `State` wird übergeben, in welchen Zustand geschaltet wird. Möglich sind die Werte `ON`, `OFF` und `TOGGLE` (Zustandswechsel).

Die Antwort des Web-IO sieht je nach Modell zum Beispiel so aus:

`output;1` (Web-IO Digital 2xIn, 2xOut)
`output;0001` (Web-IO Digital 12xIn, 12xOut bzw 12xIn, 6xRelais)

Genau wie bei der Input-Abfrage wird als Antwort das entsprechende Schlüsselwort gefolgt vom Bit-Muster der Output-Zustände gesendet. Unabhängig davon, welcher Output gesetzt wird - das Bit-Muster gibt immer den Zustand aller Outputs wieder.

Eine detaillierte Beschreibung zur Nutzung von HTTP-Requests für die Web-IO Digital Modelle finden Sie in der Request-Kommandoübersicht oder im Programmierhandbuch zum Web-IO.

Beispiele für dynamische Webseiten auf Basis von HTTP-Requests

Einfache Webseite für Upload ins Web-IO

IOs dynamisch mit JavaScript und AJAX abfragen und und zur Anzeige bringen

Mehrere Web-IOs auf einer Webseite anzeigen

Die IOs mehrerer Web-IOs mittels Cross-Origin-Reqests auf einer Seite vereinen

Web-IO mit PHP ansprechen

PHP-Script als IO-Gateway für Web-IO

Produkte

Web-IO 4.0 Digital 2xIn, 2xOut

Bei Bedarf auch über PoE zu versorgen

Web-IO 4.0 Digital 12xIn, 12xOut

12x Ausgänge (6-30V), 12x Eingänge (8-30V)

Weitere Web-IOs

Alle W&T Web-IO Digital 24V

W&T
www.WuT.de

Wir sind gerne persönlich für Sie da:

Wiesemann & Theis
GmbH
Porschestr. 12
42279 Wuppertal
Tel.: 0202/2680-110 (Mo-Fr. 8-17 Uhr)
Fax: 0202/2680-265
info@wut.de