

HTTP コマンド制御機能

HTTP 通信によりコマンドを送信することで、本製品を制御できます。
設定で本機能の「有効」・「無効」、及び HTTP 認証の設定ができます。

プロトコル	HTTP	
メソッド	GET	
リクエスト構文	http://<IP address>/api/control?<パラメータ名>=<値>[&<パラメータ名>=<値>]	
メソッド	POST	
リクエスト構文	URL : http://<IP address>/api/control リクエストヘッダ : "Content-type: application/json" データ : JSON 形式	
レスポンスデータ (制御コマンドの 場合)	Success.	制御に成功した時に返します。
	Error. [エラーコード]	制御に失敗した時に返します。

エラーコード	内容
001	対応していないメソッドです。
002	不正なパラメータ名です。
003	パラメータが指定されていません。
004	パラメータ値が指定されていません。
005	不正なパラメータ値です。

パラメータ	内容
acop=< 整数値 > [&t=< 整数値 > [&w=< 整数値 > ※[]内のパラメータ は省略可能	表示灯とブザーを制御します。 acop に続く<整数値>パラメータは、8 桁で R(赤)Y(黄)G(緑) Z(ブザー)Z(ブザー)B(青)C(白)'0' の順にパターンを指定します。 [RYGBC] : 消灯「0」、点灯「1」、点滅「2」、高速点滅「3」、変化なし「x」 [ZZ] : 消音「00」、ブザー連続音「10」、ブザー断続音「01」、変化なし「xx」 t : 継続時間 (秒) 1~65535 秒 指定した指定時間 (秒) 後に表示灯消灯・ブザー停止します。 w : 待ち時間 (秒) 1~65535 秒 指定した指定時間 (秒) 待って表示灯・ブザー制御を行います。

acop	現在の表示灯・ブザーの現在の状態を取得します。 レスポンスは 8 桁の数値で、数値の意味合いは acop による制御コマンドと同じです。
spop=< 整数値 >	指定した登録番号の音声を再生します。 spop に続く<整数値>パラメータは、8 桁で S(再生)TT(音声番号)R(リピート有無)CC(リピート回数)'00'の順にパターンを指定します。 [S] : 停止「0」、再生「1」 [TT] : 再生する音声番号「01~20」 [R] : 連続再生「0」、再生回数指定「1」 [CC] : 再生回数「01~99」
spop	現在の音声再生の状態を取得します。 レスポンスは 8 桁の数値で、数値の意味合いは spop による制御コマンドと同じです。
alof=1	表示灯を全て消灯し、音声再生・ブザーを停止します。 また、監視異常状態を解除します。
alert=< 整数値 >	表示灯とブザーを制御します。 alert に続く<整数値>パラメータは、6 桁で R(赤)Y(黄)G(緑)B(青)C(白)Z(ブザー)の順にパターンを指定します。 [RYGBC] : 消灯「0」、点灯「1」、点滅「2」、高速点滅「3」、変化なし「9」 [Z] : 消音「0」、ブザーパターン1「1」、ブザーパターン2「2」、ブザーパターン3「3」、ブザーパターン4「4」、変化なし「9」
clear=1	表示灯を全て消灯し、音声再生・ブザーを停止します。 また、監視異常状態を解除します。
led=< 整数値 >	表示灯を制御します。 alert に続く<整数値>パラメータは、5 桁で R(赤)Y(黄)G(緑)B(青)C(白)の順にパターンを指定します。 [RYGBC] : 消灯「0」、点灯「1」、点滅「2」、高速点滅「3」、変化なし「9」
sound=< 整数値 >	整数値で指定した登録番号の音声を再生します。(ワンショット) 整数値 : 1(音声 No.1) ~ 20(音声 No.20)
repeat=< 整数値 >	整数値で指定した回数リピート再生します。 整数値 : 1~99 99 を指定した場合、繰り返し連続再生します。
stop=1	再生中の音声を停止します。
logd[&n=<整数値>] [&c=<w e s>] [&h=<yes no>]	イベントログを取得します。

<p>[&s=<区切り文字>] [&o=<DESC ASC>]</p> <p>※[]内のパラメータは省略可能</p>	<p>n : ログ取得行数指定 xxx ログ取得行数(1~200)。省略時は全ログ(最大 200 行)。</p> <p>c : 文字コード指定 w : UTF-8、e : EUC-JP、s : Shift-JIS。省略時は UTF-8。</p> <p>h : ヘッダ行の有・無の指定 yes : ヘッダ行有り、no : ヘッダ行無し。省略時はヘッダ行有り。</p> <p>-s : ログ内容の区切り文字指定 区切り文字 (ハイフン(-)やアンダースコア(_)などの 1 文字)。省略時はカンマ(,)</p> <p>-o : ログの並び順 DESC : 降順、ASC : 昇順。省略時は降順。</p>
--	---

- HTTP コマンド制御機能が「無効」に設定されている場合、動作しません。
- HTTP 認証が有効の場合は、ユーザ名・パスワードが一致しないと動作しません。
- 同時に指定できるパラメータは以下の通りです。

"acop"&"spop"

"led"&"sound"

"sound"&"repeat"

"led"&"sound"&"repeat"

- alert,led は acop に、sound,stop は spop に、clear は alof に内部的に変換されて処理され、イベントログでは"ACOP"、"SPOP"、"HTTP コマンドによる全解除"というような内容で記録されます。

HTTP コマンド送信例

<acop>

「表示灯の赤を点灯、黄・緑・青・白を消灯、ブザー連続音」で動作させる場合。

- ・ HTTP メソッド GET の場合

```
http://192.168.1.1/api/control?acop=10010000
```

現在状態を取得

```
http://192.168.1.1/api/control?acop  
10010000
```

- ・ HTTP メソッド POST の場合 (curl コマンドでの送信例)

```
curl http://192.168.1.1/api/control -H "Content-type: application/json" -X POST -d  
'{"acop":"10010000"}
```

Basic 認証を行う場合 (ユーザ名=user、パスワード=password)

```
curl http://192.168.1.1/api/control --basic -u user:password -H "Content-type:  
application/json" -X POST -d '{"acop":"10010000"}
```

「表示灯の黄を点滅、赤・緑・青・白を変化なし、ブザー断続音」で動作させる場合。

- ・ HTTP メソッド GET の場合

```
http://192.168.1.1/api/control?acop=x2x01xx0
```

- ・ HTTP メソッド POST の場合 (curl コマンドでの送信例)

```
curl http://192.168.1.1/api/control -H "Content-type: application/json" -X POST -d  
'{"acop":"x2x01xx0"}
```

<spop>

No.2 の音声をワンショット再生させる場合。

- ・ HTTP メソッド GET の場合

```
http://192.168.1.1/api/control?spop=10210100
```

現在状態を取得

```
http://192.168.1.1/api/control?spop  
10210100
```

・ HTTP メソッド POST の場合 (curl コマンドでの送信例)

```
curl http://192.168.1.1/api/control -H "Content-type: application/json" -X POST -d '{"spop":"10210100"}
```

<alof>

クリア動作を実行する場合

・ HTTP メソッド GET の場合

```
http://192.168.1.1/api/control?alof=1
```

・ HTTP メソッド POST の場合 (curl コマンドでの送信例)

```
curl http://192.168.1.1/api/control -H "Content-type: application/json" -X POST -d '{"alof":"1"}
```

<alert>

「表示灯の赤を点灯、黄・緑・青・白を消灯、ブザーパターン 1」で動作させる場合。

・ HTTP メソッド GET の場合

```
http://192.168.1.1/api/control?alert=100001
```

・ HTTP メソッド POST の場合 (curl コマンドでの送信例)

```
curl http://192.168.1.1/api/control -H "Content-type: application/json" -X POST -d '{"alert":"100001"}
```

<clear>

クリア動作を実行する場合

・ HTTP メソッド GET の場合

```
http://192.168.1.1/api/control?clear=1
```

・ HTTP メソッド POST の場合 (curl コマンドでの送信例)

```
curl http://192.168.1.1/api/control -H "Content-type: application/json" -X POST -d '{"clear":"1"}
```

<led>

「表示灯の赤を点滅、黄を高速点滅、緑・青・白を点灯」で動作させる場合。

・ HTTP メソッド GET の場合

```
http://192.168.1.1/api/control?led=23111
```

・ HTTP メソッド POST の場合 (curl コマンドでの送信例)

```
curl http://192.168.1.1/api/control -H "Content-type: application/json" -X POST -d
```

```
{ "led": "23111" }
```

<sound>

No.2 の音声をワンショット再生させる場合。

- HTTP メソッド GET の場合

```
http://192.168.1.1/api/control?sound=2
```

- HTTP メソッド POST の場合 (curl コマンドでの送信例)

```
curl http://192.168.1.1/api/control -H "Content-type: application/json" -X POST -d  
{ "sound": "2" }
```

<repeat and sound>

No.1 の音声を繰り返し連続再生させる場合。

- HTTP メソッド GET の場合

```
http://192.168.1.1/api/control?repeat=99&sound=1
```

- HTTP メソッド POST の場合 (curl コマンドでの送信例)

```
curl http://192.168.1.1/api/control -H "Content-type: application/json" -X POST -d  
{ "repeat": 99, "sound": "1" }
```

<stop>

再生中の音声を停止する場合

- HTTP メソッド GET の場合

```
http://192.168.1.1/api/control?stop=1
```

- HTTP メソッド POST の場合 (curl コマンドでの送信例)

```
curl http://192.168.1.1/api/control -H "Content-type: application/json" -X POST -d  
{ "stop": "1" }
```

<led and sound>

「表示灯の赤を高速点滅、黄を点滅、緑を変化なし、白・青を消灯、No.3 音声をワンショット再生」で動作させる場合。

- HTTP メソッド GET の場合

```
http://192.168.1.1/api/control?led=32900&sound=3
```

- HTTP メソッド POST の場合 (curl コマンドでの送信例)

```
curl http://192.168.1.1/api/control -H "Content-type: application/json" -X POST -d  
{ "led": "32900", "sound": "3" }
```

<led, sound and sound>

「表示灯の赤・黄・緑を点灯、白・青を変化なし、No.10 音声を 4 回リピート (5 回再生)」
で動作させる場合。

・ HTTP メソッド GET の場合

`http://192.168.1.1/api/control?led=11199&sound=10&repeat=4`

・ HTTP メソッド POST の場合 (curl コマンドでの送信例)

`curl http://192.168.1.1/api/control -H "Content-type: application/json" -X POST -d
'{"led":11199, "sound":10, "repeat":4}'`

HTTP コマンド制御設定画面

HTTP コマンド制御に関する設定は WEB 設定ツールの詳細設定のコマンドアクセス設定にて行います。

コマンドアクセス設定

[ヘルプ](#)

RSHアクセスユーザ名	
ユーザ名1	<input type="text" value="root"/>
ユーザ名2	<input type="text" value="Administrator"/>
ユーザ名3	<input type="text" value="SYSTEM"/>
ユーザ名4	<input type="text" value="bin"/>
ユーザ名5	<input type="text" value="root"/>

RSH コマンドパスワード	
コマンドパスワード	<input type="checkbox"/> 有効

パスワード変更	
現在のパスワード	<input type="password" value="....."/>
新しいパスワード	<input type="password"/>
新しいパスワード 再入力	<input type="password"/>

HTTPコマンド制御	
HTTPコマンド制御機能	<input checked="" type="radio"/> 無効 <input type="radio"/> 有効
HTTP認証	<input checked="" type="radio"/> 認証なし <input type="radio"/> Basic認証
認証ユーザ名	<input type="text"/>
認証パスワード	<input type="password"/>

HTTP コマンド制御

HTTP コマンド制御機能	HTTP プロトコルによる表示灯・ブザー・音声のコマンド制御を使用する場合は有効にします。【デフォルト値：無効】
HTTP 認証	HTTP コマンド制御で HTTP 認証を行う場合は「Basic 認証」を選択します。【デフォルト値：認証なし】
認証ユーザ	HTTP コマンド制御で認証（Basic 認証）に使用するユーザ名を入力します（半角英数字、16 文字まで）
認証パスワード	HTTP コマンド制御で認証（Basic 認証）に使用するパスワードを入力します（半角英数字、8 文字まで）