Module erweitern

NEUER MODULPUNKT IN DER WEBOBERFLÄCHE

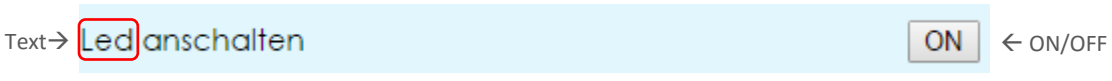
Um im Webinterface ein neues Modul anzulegen muss man auf dem Raspberry Pi in den Ordner „/var/www/html/modul“ gehen und dort einen Ordner anlegen der den Namen des neuen Moduls hat. Sobald der Ordner angelegt ist wird auch schon im Webinterface das neue Modul angezeigt.

MODULINHALT ERSTELLEN

Um nun Inhalt in das Modul zu schreiben erstellt man eine php Datei, die genau denselben Namen hat wie der Ordner. In diese kann man nun sein Programm reinschreiben. Man darf aber keinen neuen <html> oder <body> tag erstellen, da der Inhalt des Moduls in die SmartControl.php Datei implementiert wird. Man schreibt nur html, php oder javascript Ausschnitte in die Modul-Datei.

BUTTONAPI

Wenn man Knöpfe wie bei den 3 Standard Module haben möchte kann man die ButtonAPI benutzen, momentan gibt es nur An und Ausschalt Knöpfe. Aber man kann sie beliebig erweitern z.B mit einem Einstellbalken oder einem Menü.
Um einen Knopf zu erstellen gibt es die PHP-Funktion createOnButton() und createOffButton(). Es gibt jeweils 4 Übergabeparameter an die Funktion.
Beispiel:
createOnButton(“einzigartige ID“, “Text“,“auszuführende PHP-Datei beim Klick“,“Übergabe Variablen an die PHP-Datei in JSON-Format“); JSON:‘{“data1“: <inhalt>,“data2“: <inhalt>}‘}‘



Um nun die Daten in der PHP-Datei zu verarbeiten muss man aus dem Html-Post die Daten auslesen und den Inhalt dekodieren, da es ja Json Format ist. Das geht wie folgt:

$data = json_decode($_POST["DATA"]);

Nun speichert die Variable $data einen Array mit allen Übergabe Variablen. Man kann jetzt Daten einzeln in Variablen speichern mit:

$data1 = $data->{'data1'};

$data2 = $data->{'data2'};

Man kann ebenfalls den php Befehl echo benutzen, dieser wird dann in der Konsole des Browsers ausgegeben.

So sieht die JavaScript Funktion aus die aufgerufen wird, sobald der Knopf betätigt wird.

```
function sendData(pFile, pDataObject, isAsnyc) {

    var jsonc = JSON.stringify(pDataObject);
    $.ajax({
        method:     "POST",
        url:        pFile,
        data:       {DATA:pDataObject},
        cache:      false,
        async :     isAsnyc,
            success: function(msg) {
                if(msg != '') console.log(msg);
            },
            error: function(msg) {
                console.log(msg);
            }
    });
}
```

Es wird mit JQuery Ajax ein Post an den Server geschickt bzw. an die Datei, die in der Übergabevariable pFile steht. Wenn isAsync true ist kann JavaScript während der Abfrage an den Server weiterarbeiten und muss nicht warten bis eine Antwort von Server zurückkommt. In pDataObject steht der JSON String, welcher an den Server mit übergeben wird.