

# Zugriff auf die Portpins des Netzers (für IO\_base Version 1.5) über die Haussteuersoftware FHEM

[FHEM](#) ist ein in Perl geschriebener Server für Hausautomatisierung. Er lässt sich auf einigen FritzBox Modellen, Mini-computern, NAS-Systemen etc. betreiben. Zu Testzwecken ist auch ein normaler PC gut geeignet. Alles was benötigt wird, ist eine Perl-Umgebung. Beispielsweise [StrawberryPerl](#), welches auch ohne Installation genutzt werden kann. Das Einrichten von FHEM ist auf der Homepage gut erläutert.

Um den Netzer in Form von Schaltern, Schieberegler, ... in FHEM einzubinden, wird das Modul [52\\_netzer.zip](#) benötigt. Das Modul wird derzeit nicht mit FHEM ausgeliefert. Nachdem die Datei entpackt und ins Modulverzeichnis kopiert wurde, ist sie nach einem Serverneustart nutzbar.

## Vorbereitung des Netzers

Zuerst müssen über das Webinterface ([io](#)) des Netzers die Port so konfiguriert werden, wie man sie später nutzen möchte. Anschließend wird der [gpioserver](#) aktiviert. Über Diesen kommunizieren FHEM und der Netzer miteinander. Wenn ein Port mit Ereignistrigger konfiguriert ist dann werden bei jedem Ereignis alle Portwerte in FHEM aktualisiert.

## Konfigurieren von FHEM

Folgende Zeilen werden in die `fhem.cfg` eingetragen und nach dem speichern über die FHEM Website sofort übernommen. Als erstes wird ein neues Netzermodul definiert:

```
define <name> Netzer <host:port>
```

Bereits jetzt sind alle Ports als Eingang verfügbar. Sollen sie einer anderen Funktion dienen so werden die gemäß der Einstellung der Netzer Weboberfläche auch in der `fhem.cfg` eingetragen:

```
attr <name> Port_a <out|PWM|ADC|cnt|in>
```

- out: Ausgang
- PWM: PWM Ausgang (die set Funktion wird mit einem Schieberegler versehen)
- ADC: A/D Konverter Eingang
- cnt: digitaler Eingang, es wird ein zusätzliches Reading mit dem Zählerwert angelegt. Dieses lässt sich lesen und schreiben
- in: Eingang (in diesem Fall kann das Attribut auch weggelassen werden)

Jetzt ist die Konfiguration fertig. Um das ganze schöner darzustellen lassen sich die einzelnen Readings der Ports mit der "readingsProxy" Funktion in FHEM auch einzeln darstellen.

## Beispielkonfiguration

```
define netzer01 Netzer 192.168.178.33:65000
attr netzer01 Port_a out
attr netzer01 Port_d PWM
attr netzer01 Port_e ADC
attr netzer01 Port_f in
attr netzer01 room Netzer

define netzer1a readingsProxy netzer01:Port_a
attr netzer1a room Netzer
attr netzer1a setFn {($CMD eq "off")?"a 0":"a 1"}
attr netzer1a setList on off
attr netzer1a valueFn {($VALUE == 0)"off":"on"}

define netzer1b readingsProxy netzer01:Port_b
attr netzer1b room Netzer
attr netzer1b setFn {($CMD eq "off")?"b 0":"b 1"}
attr netzer1b setList on off
attr netzer1b valueFn {($VALUE == 0)"off":"on"}
define act_netzer1b notify netzer1b {if ("% eq "off") {fhem ("get
test03");;}}

define netzer1d readingsProxy netzer01:Port_d
attr netzer1d devStateIcon 0:off 100:on \d:dim06% 1\d.*:dim12% 2\d.*:dim25%
3\d.*:dim37% 4\d.*:dim43% 5\d.*:dim50% 6\d.*:dim62% 7\d.*:dim75%
8\d.*:dim87% 9\d.*:dim93%
attr netzer1d room Netzer
attr netzer1d setFn {($CMD eq "off")?"d 0":($CMD eq "on")?"d 1000":"d " .
$CMD *10}
attr netzer1d setList off:noArg on:noArg state:slider,0,1,100
attr netzer1d valueFn {($VALUE == 0)"off":($VALUE / 10)}
attr netzer1d webCmd off:state:on
```

From:

<http://mobacon.de/dokuwiki/> - **MoBaCon**

Permanent link:

<http://mobacon.de/dokuwiki/doku.php?id=de:netzer:fhem&rev=1391855255>

Last update: **2025/06/11 20:42**

