ACHTUNG! Diese Seite wird nicht mehr länger gepflegt.
Wechselt auf die englischsprachige Seite für eine aktuelle
Anleitung (Link unten links).

Der GPIO-Server

Ab Version 1.5 ist der GPIO-Server ausschließlich im **IO_base** Projekt implementiert. Für **IO_pro** existiert mit der Kommandoschnittstelle eine mächtigere Alternative.

Einstellungen

Falls der Server aktiviert ist, öffnet der Netzer nach dem Booten einen TCP Port (Werkseinstellung 65000). Die Aktivierung und der Port kann über die Webseite mit allgemeinen Einstellungen vorgenommen werden (siehe Abbildung). Die Einstellungen werden direkt wirksam.

Für einfache Tests des Servers und des Protokolls kann jedes Terminalprogramm verwendet werden, was Verbindungen über Raw TCP Sockets aufbauen kann. Dazu zählt z.B. das Windows HyperTerminal. Nach erfolgreichen Verbindungsaufbau erfolgt die Authentifikation, falls aktiviert. Sonst erfolgt der direkte Eintritt in die Protokollphase.

Protokoll

Im GPIO-Server ist ein einfaches Protokoll implementiert.

Damit können digitale Ein- und Ausgänge sowie ADC und PWM/Impuls-Ausgänge gelesen werden. Es können nur digitale Ausgänge sowie PWM/Impuls-Ausgänge geschrieben werden. Außerdem dürfen diese Ausgänge nicht durch Funktionen des seriellen Servers belegt sein.

Gelesen oder geschrieben wird mittels Kommandos. Einzelne Kommandos sind voneinander mit Whitespaces (Leerzeichen, Tab, Zeilenumbruch, etc. - alle Zeichen mit dem ASCII-Code ⇐ 32) getrennt.

Jeder Pin ist mittels seiner ID eindeutig adressierbar. Hier eine Übersicht über alle IDs:

Name	IO0	IO1	IO2	IO3	IO4	IO5	TX	RX	SPI_CS	SPI_INT	SPI_CLK	SPI_MI	SPI_MO	Alle Pins
ID	a	b	c	d	e	f	g	h	i	j	k	l	m	x

Das Lesen von Pins funktioniert wie folgt:

```
a=? Serverantwort: a=1
e=? Serverantwort: e=03ff
m=? Serverantwort: m=0
x=? Serverantwort: x=0011
```

Je nach dem, ob der Pin digital oder als ADC/PWM konfiguriert ist, liefert der GPIO-Server binäre oder numerische Werte.

Die Joker-ID **x** liefert immer den digitalen Zustand aller Pins, unabhängig davon, ob ein Pin binäre oder numerische Werte liefert. Auf Bitebene ergeben sich die Zustände der einzelnen Pins wie in der folgenden Tabelle:

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	m	l	k	j	i	h	g	f	e	d	c	b	a

Das Schreiben auf Pins erfolgt analog:

```
a=1 Serverantwort: a=1
d=0123 Serverantwort: d=0123
m=0 Serverantwort: m=0
x=0123 Serverantwort: x=0123
```

Die Joker-ID **x** beschreibt nur die Pins, die als digitale Ausgänge konfiguriert sind und nicht durch den seriellen Server belegt sind.

Events

Für jeden digitalen Eingang kann ein Ereignistrigger auf der GPIO-Konfigurationsseite aktiviert werden.

Damit sendet der GPIO-Server selbständig Protokollmeldungen bei entsprechend konfigurierter

Änderung.

Flankenzähler

Für die IO-Kanäle a, b und c sind Flankenzähler implementiert, die über die spezielle Protokoll-ID **z** ausgelesen werden können.

Das Auslesen der Zähler funktioniert wie folgt:

```
za=? Serverantwort: za=0001
zb=? Serverantwort: zb=0000
zc=? Serverantwort: zc=0123
```

Zählerwerte werden hexadezimal mit einer Breite von 16 Bit übermittelt. Der Zählerwert selbst ist in den unteren 15 Bit gespeichert. Das MSB des Zählers ist das Carry-Bit, welches einen Überlauf bzw. Unterlauf des Zählers anzeigt. Ist dieses einmal gesetzt, bleibt es stehen, bis es manuell zurückgesetzt wird.

Auch das Schreiben der Zählerwerte ist möglich:

```
za=0 Serverantwort: za=0000
```

Bei Änderung des Zählerwertes wird dieser nicht automatisch übermittelt, er muss periodisch abgefragt werden.