

PIO84-OCO

14. Juli 2011



Inhaltsverzeichnis

1	Eigenschaften	5
2	Bedienung	6
2.1	Die Weboberfläche	6
2.2	Benutzerspezifische Webseite	11
2.3	Steuerung per Skript	12
2.4	Signalgeber	16
2.5	Fernsteuerung über Remote Skripte	17
2.6	Fernsteuerung per MAD Bridge	20
3	Schnittstellen	21
3.1	Softwareschnittstellen	21
3.2	Hardwareschnittstellen	21
3.3	Stromversorgung	21
3.4	Steckerbelegung D-SUB15	23
3.5	Ein- und Ausgangsschaltungen	24
4	Technische Daten	25
4.1	Elektrische Daten	25
4.2	Umgebungsbedingungen	25
4.3	Werkseinstellungen	25
4.4	Software	26
5	Das Demoboard	26
6	Dokument Historie	27

Abbildungsverzeichnis

1	Ein einfacher Blinker	12
2	Noch ein Laufflicht.	12
3	Einlesen eines Eingangs.	14
4	Einlesen aller Eingänge.	14
5	Acht gedimmte LEDs.	15

6	Ein kurzer Piep mit 1KHz mit voller Lautstärke, 100ms lang.	16
7	Ein abklingendes, akustisches Signal.	16
8	Kommandoorientierte Fernsteuerung	18
9	Interaktive Fernsteuerung	18
10	Schnittstellen	21
11	Innenschaltung Variante3.	24
12	Innenschaltung Variante3.	24

Tabellenverzeichnis

1	Steckerbelegung Stromversorgung.	21
2	Steckerbelegung DSUB15.	23
3	Elektrische Daten	25
4	Umgebungsbedingungen	25
5	Auslieferungszustand	25
6	Installierte Softwarekomponenten	26

Hinweis

Die Angaben in diesem Datenblatt sind von uns auf Übereinstimmung mit der beschriebenen Hardware bzw. Software geprüft worden.

Aus verschiedenen Gründen sind Abweichungen dennoch nicht ausgeschlossen, weswegen wir für eine vollständige Übereinstimmung keine Gewährleistung übernehmen können.

Alle zukünftigen, relevanten Änderungen werden im Abschnitt 6 dokumentiert.

Für Anregungen oder Verbesserungsvorschläge sind wir jederzeit aufgeschlossen.

1 Eigenschaften

Das NetPort PIO84OCO ist eine universelle Steuerung, die durch den Endanwender auf viele Bedürfnisse angepasst werden kann. Der Endanwender wird in die Lage versetzt modernste Webtechnologie mit der Steuerung von externer Hardware zu verbinden.

Ein integrierter Skriptinterpreter ermöglicht die Implementierung komplexer, autarker Steuerungs- und Regelungsaufgaben. Ein Webserver bietet die Möglichkeit eigene Webseiten anzuzeigen, die in Interaktion mit der Hardware stehen können. Vielfältige Konfigurationsmöglichkeiten lassen die Anwendung im Industriellen- und Konsumerbereich zu. Auf der Hardwareebene kann das Modul einfach um zusätzliche Komponenten erweitert werden.

Mögliche Anwendungen sind:

- Steuerungen aller Art.
- Fernsteuerung von Aktoren über ein Webinterface.
- Steuerung von Modellbauservos per Ethernet.
- Messung von physikalischen Grössen mit Webinterface.
- Inbetriebnahme- und Test bei der Entwicklung elektronischer Komponenten.

Das PIO84 OCO verfügt über **acht** Open Kollektorausgänge, die für einen maximalen Strom I_{OX} (Siehe Tabelle 3) ausgelegt sind. Weiterhin sind **vier** optisch entkoppelte Eingänge vorhanden. Die Fotodioden der Optokoppler benötigen einen Strom I_{IX} zum sicheren Schalten.

Weitere Merkmale sind:

- Piezo Signalgeber
Frequenz im Bereich $f_{PiezoMin}$ - $f_{PiezoMax}$ per Software einstellbar. Lautstärke per Software einstellbar. (0..100%).
- Ethernet Schnittstelle
- USB Schnittstelle
- DSUB15 Verbinder für Ein- und Ausgänge
- **C Interpreter**
(Bis auf wenige Ausnahmen ISO C Kompatibel. Zusätzlich werden einige C++ Konstrukte unterstützt.)

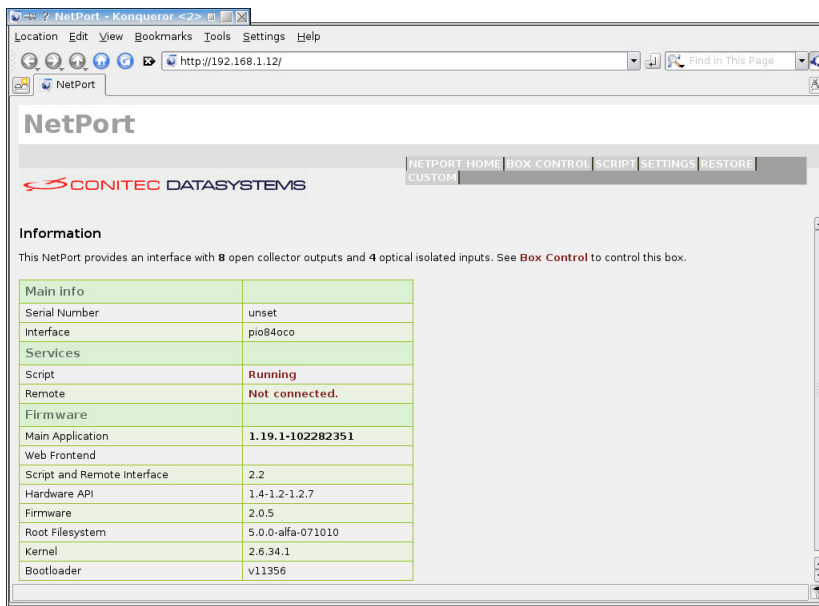
2 Bedienung

Das PIO Modul kann auf verschiedene Arten verwendet werden. Siehe Softwareschnittstellen (3.1).

2.1 Die Weboberfläche

Sie erreichen das Modul, indem Sie dessen IP-Adresse in Ihren Browser eingeben. Zum Beispiel (Werkseinstellung):

<http://192.168.1.12>



Kurz darauf erscheint die Webseite.

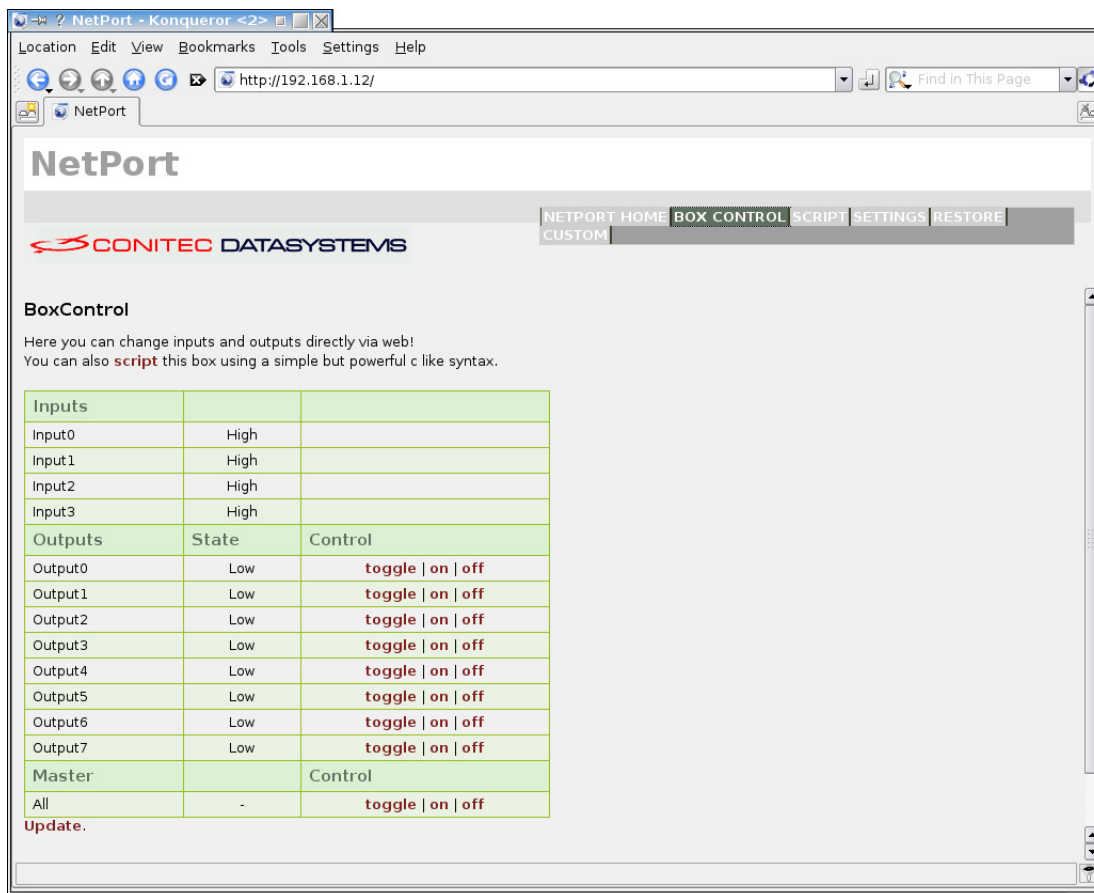
2.1.1 BOX CONTROL

Auf dieser Seite können Sie die Ausgänge kontrollieren und die Eingänge einlesen.

Sie können die Webseite aktualisieren, indem Sie "Update" betätigen.

Hinweis

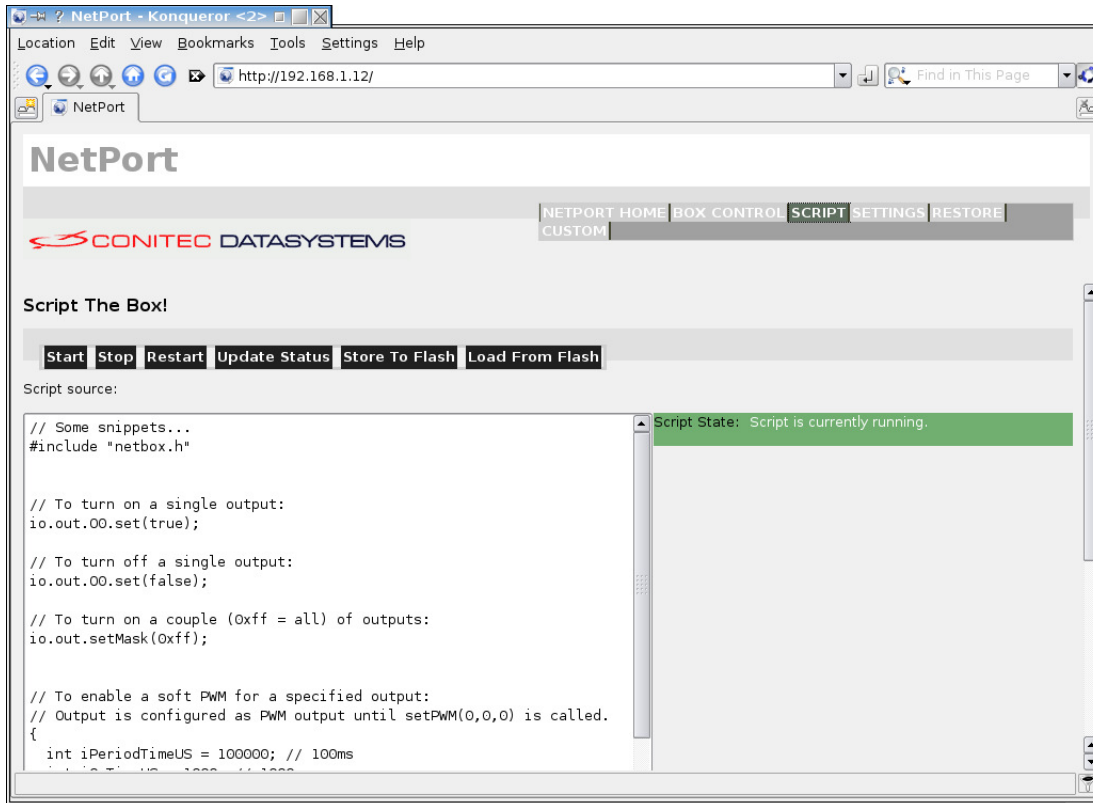
Die Eingänge sind mit **Input0..Input3** bezeichnet. Die Ausgänge entsprechend **Output0..Output7**. Die Namen können auch geändert werden. (Siehe Einstellungen 2.1.3.)



Bitte beachten Sie, dass die manuelle Steuerung der Box auch dann erlaubt ist, wenn bereits ein Skript auf der Box ausgeführt- oder die Box anderweitig ferngesteuert wird.

2.1.2 SCRIPT

In diesem Bereich der Webseite können Sie einfache C-Skripte auf die Box laden und ausführen. Die Skripte können dazu dienen, einfache Abläufe zu realisieren oder auch komplexe Steuerungsaufgaben zu erledigen. Skripte können direkt per Weboberfläche eingegeben und gespeichert werden, so dass sie beim nächsten Start automatisch ausgeführt werden. (Siehe Einstellungen 2.1.3).



Syntax- oder Laufzeitfehler werden ebenfalls in der Weboberfläche dargestellt.

2.1.3 SETTINGS (Einstellungen)

Hier können die Einstellungen der Box verändert werden. Beachten Sie, dass neue Einstellungen erst wirksam werden, wenn “Apply Settings” betätigt wird.

2.1.3.1 Autostart

Diese Funktion bezieht sich auf das Skript. Wenn sie Aktiv ist wird das Skript sofort nach dem Einschalten ausgeführt.

2.1.3.2 When script fails : Restart

Wenn diese Funktion eingeschaltet ist, wird das Skript nach auftreten eines Fehlers sofort neu gestartet. (Fehler können Laufzeitfehler oder auch Syntaxfehler sein.)

Diese Option ist bei Auslieferung eingeschaltet. Für die Entwicklung und zum Test von Skripten sollte diese Funktion ausgeschaltet sein.

Hinweis

2.1.3.3 When script fails: Warning Beep

Wenn diese Funktion eingeschaltet ist, ertönt ein akustisches Signal, sobald ein Skriptfehler aufgetreten ist.

2.1.3.4 When script terminates: Restart script automatically.

Wird das Skript regulär beendet und diese Option ist eingeschaltet wird es automatisch neu gestartet.

2.1.3.5 Output Names

Hier können Namen für die Ausgänge eingetragen werden. Weiterhin ein Symbol zur Kennzeichnung für einen Aktiven Ausgang (“Name for on”), sowie eine Bezeichnung für einen passiven Ausgang (“Name for off”). Das Feld “Negated” markiert einen invertierten Ausgang.

2.1.3.6 Input Names

Analog zu den Bezeichnungen der Ausgänge können hier Bezeichnungen der Eingänge eingetragen werden. Auch die Eingänge können negiert werden.

2.1.3.7 Negierte Ein- und Ausgänge

Bei einem negierten Ein- oder Ausgang erscheint auf der Box-Control Seite ein Negationsstrich über der Bezeichnung. Die Logik ist in diesem Fall negiert.

Ein negierter Eingang ist **true**, wenn dessen Optokoppler **nicht angesteuert** wird. Ein negierter Ausgang ist **aktiv**, wenn der Ausgang auf **false** gesetzt wird.

Die Negation betrifft die Steuerung per Webseite, das Skript (nachdem es neu gestartet wurde) und auch die Fernsteuerung.

2.1.4 Einstellungen

Ab Firmware Version 1.14 wird die Funktion “Restore” zur Verfügung gestellt. Diese Funktion ermöglicht es dem Benutzer alle Einstellungen der Box als Datei zu sichern und später zurückzuschreiben.

2.1.4.1 Einstellungen speichern

Um die Einstellungen zu sichern genügt ein Klick auf “SETTINGS” > “Download Configuration”. Je nach Browser wird die Datei sofort heruntergeladen oder es kommt eine Aufforderung zum Speichern der Datei. Die Datei mit der Endung “.netbox” enthält alle Einstellungen wie IP-Adresse, I/O Konfiguration und auch das Skript. Zusätzlich wird die Seriennummer des PIO Moduls gespeichert.

2.1.4.2 Gespeicherte Einstellungen zurückschreiben

Das Zurücksichern der Einstellungen geschieht über die Weboberfläche im Menüpunkt “RESTORE”. Hier kann angegeben werden, welche Inhalte wiederhergestellt werden sollen:

- “Restore Script”
Das aktuelle Skript wird das in der Konfigurationsdatei (.netbox) gespeicherte Skript ersetzt.
- “Restore Settings”
IP Adresse, I/O Konfiguration und alles, was sich unter “SETTINGS” befindet wird durch die Konfigurationsdatei ersetzt.
- “Restore only when serial number matches”
Wenn diese Option angeklickt ist wird zunächst geprüft, ob die Seriennummer des aktuellen Gerätes mit der Seriennummer der Konfigurationsdatei übereinstimmen. Nur wenn das der Fall ist werden Änderungen vorgenommen. Sollten die Seriennummern **nicht** übereinstimmen wird eine Fehlermeldung ausgegeben und **nichts** geändert.
- “Reboot after restoring data”
Wenn diese Option aktiv ist, wird die Box nach Anwendung der Konfiguration automatisch neu gestartet.

Sobald Sie die Datei auf der Webseite angegeben haben und “Restore” drücken werden die Einstellungen übernommen und die Box ggf. neu gestartet.

2.2 Benutzerspezifische Webseite

Das Modul erlaubt es dem Anwender eine eigene Webseite zu definieren und auf dem Modul abzulegen. Diese Webseite ("Custom Pages") erscheint je nach Einstellung sofort auf der IP Adresse der Box oder im Bereich "CUSTOM" auf der Hauptseite.

2.2.1 Möglichkeiten

Die benutzerspezifische Webseite erlaubt es mit Hilfe des Netports eine vollständige, webgesteuerte Steuerung zu implementieren. Die Webseite kann dabei aus mehreren Dateien bestehen. (Bilder, StyleSheets, JavaScript, ...)

Mit Hilfe definierter Schnittstellen kann über JavaScript auf die I/O Ports und die Einstellungen zugegriffen werden. Weiterhin ist ein Datenaustausch zwischen JavaScript und dem Box-Skript möglich. Weiteres dazu im Dokument "netPortWeb.pdf".

2.2.2 Grenzen

Das Netportmodul stellt für die Webseite einen Speicherplatz von 256kB zur Verfügung.

2.2.3 Entwickeln der Benutzerspezifischen Webseite

Die Benutzerspezifische Webseite wird beim Systemstart in das FTP Verzeichnis unter "custom-html" geladen. Sofern sich dort eine Datei mit dem Namen "index.html" befindet und die Einstellung "Show Custom Pages as default" gesetzt ist, wird diese Datei Hauptseite verwendet.

Sie können Ihre eigene Webseite mit Hilfe eines FTP Clients auf das Netport Modul laden.

2.2.4 Speichern der Webseite

Im "SETTINGS" Bereich befindet sich ein Schalter "Store Custom Website". Drücken Sie diesen Schalter, um alle Dateien im "custom-html" Verzeichnis dauerhaft zu speichern.

Die aktuell gespeicherte Webseite wird dadurch unwiederbringlich ersetzt. Machen Sie sich ggf. Sicherheitskopien des "custom-html" Verzeichnisses.

Hinweis

2.3 Steuerung per Skript

In diesem Kapitel finden Sie einige Skriptbeispiele. Sie können Diese Skriptbeispiele heraus kopieren und direkt in der Weboberfläche Testen.

2.3.1 Steuerung der Ausgänge

2.3.1.1 Blinker

```
#include "netbox.h"

while (true)
{
    io.set(0, true);
    pause(100);
    io.set(0, false);
    pause(200);
}
```

Quelle: ./snippets/blink.msl

Abbildung 1: Ein einfacher Blinker

Dieses Beispiel lässt den Ausgang O0 blinken. (100ms an, 200ms aus).

2.3.1.2 Lauflicht

```
#include "netbox.h"

int iDir = 1;
int iIdx = 0;
while (true)
{
    iIdx += iDir;
    if ((iIdx >= 7) || (iIdx == 0))
        iDir *= -1;
    io.out.setMask(1 << iIdx);
    pause(100);
}
```

Quelle: ./snippets/lauflicht2.msl

Abbildung 2: Noch ein Lauflicht.

Es werden nacheinander alle LEDs eingeschaltet. (Wandernder Punkt.)

2.3.2 Einlesen der Eingänge

Wie die Ausgänge können auch die Eingänge einzeln oder als Maske eingelesen werden.

```
#include "netbox.h"

// All off.
io.out.setMask(0);

while (true)
{
    bool bo = io.get(0);
    io.set(0, bo);
}

Quelle: ./snippets/input1.msl
```

Abbildung 3: Einlesen eines Eingangs.

Dieses Beispiel setzt O0 in Anhängigkeit des ersten Eingangs. (Kopiert das Signal...)

2.3.3 Einlesen aller Eingänge

```
#include "netbox.h"

// All off.
io.out.setMask(0);

enum Switches
{
    sw1 = (1 << 0),
    sw2 = (1 << 1),
};

while (true)
{
    bool bo = io.in.getMask() & (sw1 | sw2);
    io.out.O0.set(bo);
}

Quelle: ./snippets/input2.msl
```

Abbildung 4: Einlesen aller Eingänge.

2.3.4 Erzeugung von PWM Signalen

Die Ausgänge können als PWM Generatoren verwendet werden. Ein Kernetreiber erzeugt dabei bis zu acht unabhängige Soft-PWM Signale, die frei per Skript oder auch per Fernsteuerung konfigurierbar sind. Ein Ausgang ist so lange als PWM Ausgang aktiv, bis die PWM explizit abgeschaltet- oder das Skript neu gestartet wird.

```
#include "netbox.h"

// 1ms = 1000us
const int us = 1;
const int ms = 1000;

io.setPWM(0, 15*ms, 100*us);
io.setPWM(1, 15*ms, 200*us);
io.setPWM(2, 15*ms, 400*us);
io.setPWM(3, 15*ms, 1*ms);
io.setPWM(4, 15*ms, 2*ms);
io.setPWM(5, 15*ms, 3*ms);
io.setPWM(6, 15*ms, 4*ms);
io.setPWM(7, 15*ms, 5*ms);

// wait forever...
while (true)
    pause();
```

Quelle: ./snippets/pwm1.msl

Abbildung 5: Acht gedimmte LEDs.

Wenn das Modul mit dem Demoboard verbunden ist kann man acht gedimmte LEDs erkennen.

2.3.5 Ausschalten der PWM Funktion

Der PWM Modus wird verlassen, sobald entweder die Frequenz oder die On-Zeit auf Null gesetzt werden.

```
#include "netbox.h"

io.setPWM(0, 0,0); // disable...
```

Quelle: ./snippets/pwmOff.msl

2.4 Signalgeber

2.4.1 Ein einfaches Signal

Um einen Ton zu erzeugen:

```
#include "netbox.h"

mcu.piezo.beep(1000, 100);
pause(200);
mcu.piezo.off();
// Wait forever...
while (true) pause();

Quelle: ./snippets/piezo1.msl
```

Abbildung 6: Ein kurzer Piep mit 1KHz mit voller Lautstärke, 100ms lang.

2.4.2 Ein abklingendes “Ding”

```
#include "netbox.h"

for (int i = 0; i < 100; ++i)
{
    mcu.piezo.beep(1000, 100-i);
    pause(1);
}
mcu.piezo.off();
// Wait forever...
while (true) pause();

Quelle: ./snippets/piezo2.msl
```

Abbildung 7: Ein abklingendes, akustisches Signal.

2.5 Fernsteuerung über Remote Skripte

Ab Firmware Version 1.14 erlaubt die NetBox die Ausführung von Skripten, die per Netzwerkverbindung übertragen wurden. Dieses Merkmal ist zur einfachen Fernsteuerung der Box vorgesehen.

Die Box bedient hierzu den TCP Port (1233). Ein- oder mehrere Clients können sich zu diesem Port verbinden, ein Skript hinsenden und somit die Box fernsteuern. Dabei können prinzipiell die gleichen Skripte verwendet werden, wie sie auch im vorherigen Kapitel gezeigt wurden - mit zwei Ausnahmen:

1. `#include <netbox.h>` ist nicht notwendig und kann weggelassen werden.
2. Die Endlosschleife im unteren Teil des Programms kann / sollte weggelassen werden, damit das Kommando beendet werden kann.

Alle Skripte (Remote Skripte und das Box Skript) werden quasiparallel ausgeführt. Wenn sich ale mehrere Clients verbinden können also auch mehrere Kommandos (z.B. Berechnungen) parallel zueinander ausgeführt werden. Jedes Skript hat vollen Zugriff auf alle Ressourcen wie I/O Leitungen.

Die Fernsteuerung über Remote Skripte ist nur aktiv, wenn unter Settings "Allow remote script" aktiv ist.

Hinweis

2.5.1 Ablauf einer Fernsteuerungssitzung

Es gibt zwei Möglichkeiten einer Fernsteuerungssitzung:

1. Kommandoorientiert

Hierbei wird nur ein Kommando (oder eine kurze Sequenz) an die Box gesendet. Die Verbindung wird durch die Box geschlossen.

2. Interaktiv

Es können nacheinander mehrere Kommandos (oder Sequenzen) gesendet werden. Die Verbindung wird durch den Client beendet.

Die folgenden Abbildungen verdeutlichen das Vorgehen. (Es wurde exemplarisch die IP Adresse 192.168.1.12 gewählt.)



Abbildung 8: Kommandoorientierte Fernsteuerung

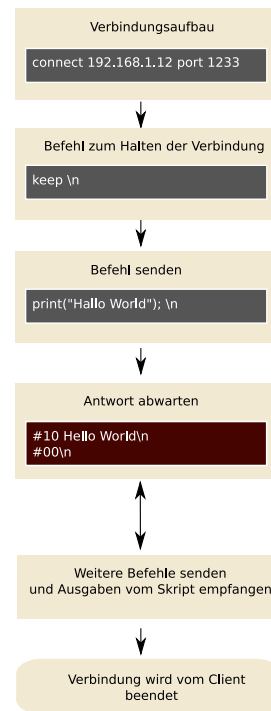


Abbildung 9: Interaktive Fernsteuerung

2.5.2 Beispiele für Fernsteuerskripte

In diesem Abschnitt finden Sie Beispiele für einfache Fernsteuerskripte. Sie können diese Skripte mit Hilfe eines TCP Sockets direkt an die Box senden. Alternativ können Sie auch das Programm “telnet” verwenden:

```
telnet 192.168.1.12 1233
```

Quelle: ./snippets-remote/telnet.msl

Alle Eingaben die Sie jetzt machen, werden direkt an die NetBox übertragen und bei drücken der Return Taste ausgeführt. Je nachdem, ob Sie als erstes “keep” eingeben wird die Verbindung nach Ausführung des Kommandos geschlossen - oder offen gehalten.

2.5.2.1 Eine I/O Leitung einschalten

```
io.out.00.set(true);
```

Quelle: ./snippets-remote/remote-gpio-on.msl

2.5.2.2 Den Zustand einer Leitung einlesen

```
print(io.out.I0.get());
```

Quelle: ./snippets-remote/remote-gpio-in.msl

2.5.2.3 Piezo einschalten

```
mcu.piezo.beep(1000, 100); // 1kHz, 100% Vol.
```

Quelle: ./snippets-remote/remote-beep-on.msl

2.5.2.4 Piezo ausschalten

```
mcu.piezo.off();
```

Quelle: ./snippets-remote/remote-beep-off.msl

2.5.2.5 Ein kurzer Piep

```
mcu.piezo.beep(1000, 100); // 1000Hz, 100% Lautstärke  
pause(100); // 100ms warten
```

```
mcu.piezo.off();
```

Quelle: ./snippets-remote/remote-beep.msl

2.6 Fernsteuerung per MAD Bridge

Die Box lässt sich z.B. mit Hilfe der GalepX Software fernsteuern. Ein Beispiel, wie eine solche Fernsteuerung aussehen kann findet sich in der GalepX Distribution unter skripts/pio84oco/pio84oco.gxs. Dieses Skript stellt ein Fenster dar, welches die Eingangszustände des PIO Moduls darstellt und eine Steuerung der Ausgänge über Checkboxes erlaubt.

3 Schnittstellen

3.1 Softwareschnittstellen

Das Modul kann auf verschiedene Weise gesteuert werden:

- Steuerung per Weboberfläche
Siehe 2.1.
- Steuerung per Java Script (AJAX)
Dokument: **NetportWeb.pdf**.
- Steuerung mit eingebautem C Skript
Siehe 2.3.
- Fernsteuerung mit Remote Skripten
Siehe 2.5.
- Fernsteuerung mit- oder von einem anderen Modul (Bridge)
Dokument: **NetportBridge.pdf**.

3.2 Hardwareschnittstellen

Schnittstelle		Voreinstellung	Dienste
Ethernet	10/100MBPs, Auto Negotiation 00:11:ef:e0:xx:xx bis 00:11:ef:e7:xx:xx	192.168.1.12	FTP, Telnet, Http, MADBridge
USB-Device	Fullspeed (12 MBits/s), Ethernet Emulation, Powered by Device	192.168.167.12	FTP, Telnet, Http, MADBridge
Stromversorgung	3,81mm, 2 Polig		

Abbildung 10: Schnittstellen

3.3 Stromversorgung

Pin	Verwendung	Anmerkung
1	UIn	Maximal 28V
2	GND	

Tabelle 1: Steckerbelegung Stromversorgung.

Der Stromversorgungseingang ist verpolungssicher.



Das Modul kann sowohl über die Eingangsspannungsbuchse, als auch über **Pin 8**(UIn) und **Pin 4**(GND) versorgt werden.

Hinweis

3.4 Steckerbelegung D-SUB15

Das Modul bietet einen DSUB15 Stecker (**male**).

Pin	Verwendung	Anmerkung
1	IN-GND	Gemeinsame Kathode (Optokoppler)
2	IN3	Eingang
3	IN2	Eingang
4	GND	GND für Ausgänge, sowie Versorgung
5	IN1	Eingang
6	IN0	Eingang
7	O7	Open Kollektor Ausgang
8	UIN	Versorgungsspannung (Siehe Hinweis)
9	O6	Open Kollektor Ausgang
10	O5	Open Kollektor Ausgang
11	O4	Open Kollektor Ausgang
12	O3	Open Kollektor Ausgang
13	O2	Open Kollektor Ausgang
14	O1	Open Kollektor Ausgang
15	O0	Open Kollektor Ausgang

Tabelle 2: Steckerbelegung DSUB15.

3.5 Ein- und Ausgangsschaltungen

3.5.1 Ausgänge mit offenem Kollektor

Diese Beschaltung betrifft **O0-O7**. Hierbei handelt es sich um Open Collector Ausgänge, die wie folgt implementiert sind:

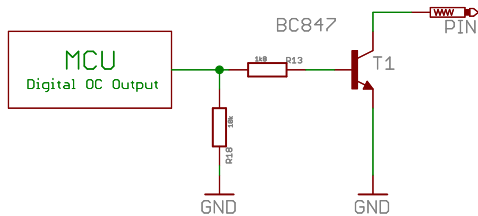


Abbildung 11: Innenschaltung Variante3.

3.5.2 Eingänge

Diese Beschaltung betrifft **I0-O3**. Es handelt sich um optisch entkoppelte Eingänge.

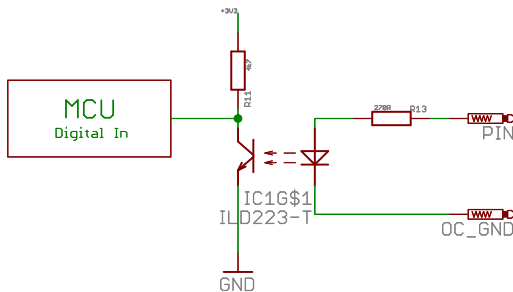


Abbildung 12: Innenschaltung Variante3.

4 Technische Daten

4.1 Elektrische Daten

Parameter	Min	Typ	Max	Einheit
Versorgungsspannung	8	12	28V	V (DC)
Leistungsaufnahme		1		W
Stromaufnahme USB		<20		μA
I_{OX}			100	mA
I_{IX}	5		20	mA
f_{Piezo}	800		4000	Hz
Zul. Betriebstemperatur	0		60	°C
Lagertemperatur	-25		75	°C

Tabelle 3: Elektrische Daten

4.2 Umgebungsbedingungen

Parameter	Min	Typ	Max	Einheit
Zul. Betriebstemperatur	0		60	°C
Lagertemperatur	-25		75	°C
Schutzart	-	-	-	IP20

Tabelle 4: Umgebungsbedingungen

4.3 Werkseinstellungen

Die folgenden Daten entsprechen den Daten bei Auslieferung des Moduls.

Parameter	Wert
IP-Adresse (Ethernet)	192.168.1.12
IP-Adresse (USB)	192.168.167.12

Tabelle 5: Auslieferungszustand

4.4 Software

Komponente	Wert	Version	Lizenz
First Stage Loader	LdDataFlash	svn-201	Proprietär / Conitec
Second Stage Loader	LdLinux++	svn-201	Proprietär / Conitec
Betriebssystem	Linux	2.6.34-1	GPL
Webserver	MADWeb	1.1.2.2	Proprietär
FTP Server	BusyBox	1.17.0	GPL
Basissystem	BusyBox	1.17.0	GPL
Konfiguration	dfConfig	1.7.2	Proprietär / Conitec

Tabelle 6: Installierte Softwarekomponenten

Die Versionen der einzelnen Komponenten werden auf der Web-Hauptseite des Moduls dargestellt.

5 Das Demoboard

Das PIO84 Demoboard erlaubt den einfachen Einstieg in die Arbeit mit dem Modul. Es verfügt über acht LEDs, welche die Ausgangszustände anzeigen und vier Taster mit denen die Eingänge stimuliert werden können.

Je nach Bestückungsoption können zusätzlich Leistungsschalter, sowie eine schaltbare Spannungsquelle vorhanden sein.

6 Dokument Historie

- Revision 1.
Initial.
- Revision 1.1
 - Fixed: Some typos.
- Revision 1.2
 - Hinzugefügt: Einstellungen Sichern / Restore
 - Hinzugefügt: Fernsteuerung über Remote-Skripte
- Revision 1.3
 - Hinzugefügt: Custom Pages.
 - Vereinfacht: Beispiele.
 - Hinzugefügt: Ein- und Ausgangsschaltungen.