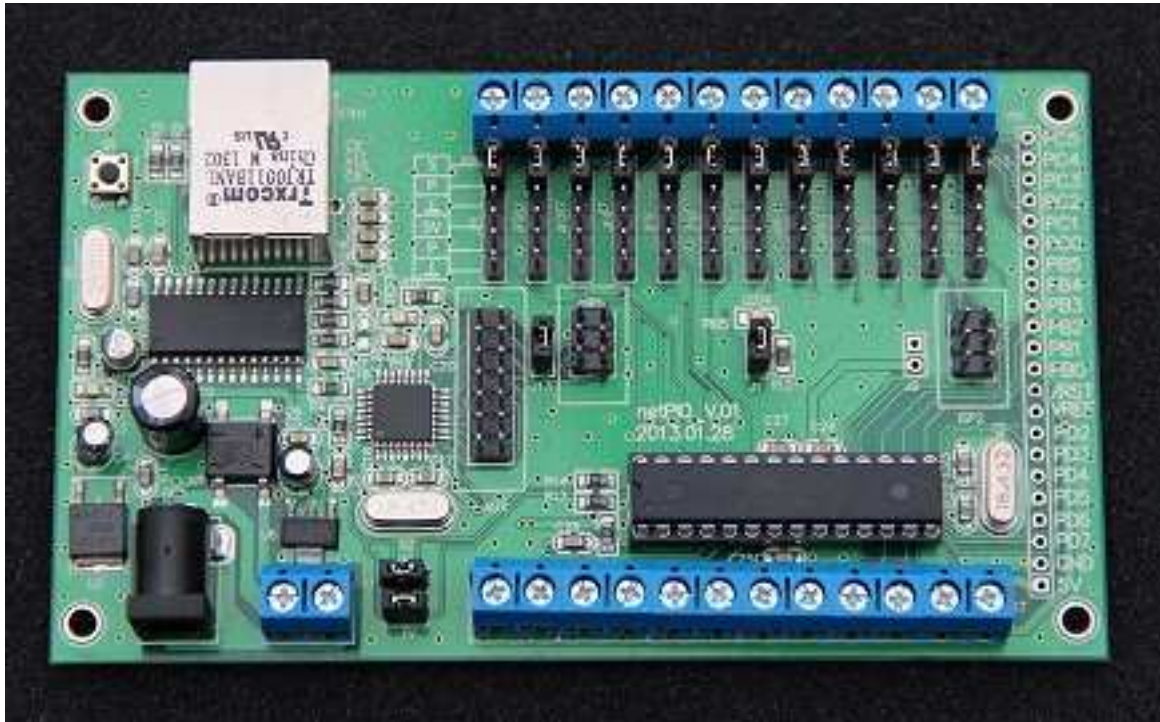


# ABACOM - netPIO



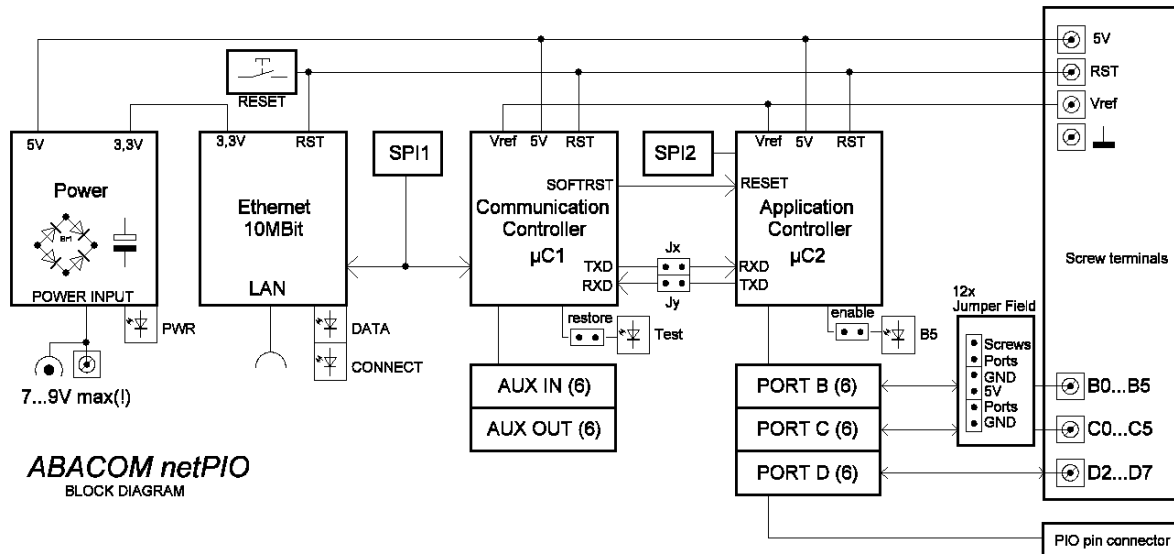
## Download

[http://www.abacom-online.de/div/setup\\_netPIO.exe](http://www.abacom-online.de/div/setup_netPIO.exe)

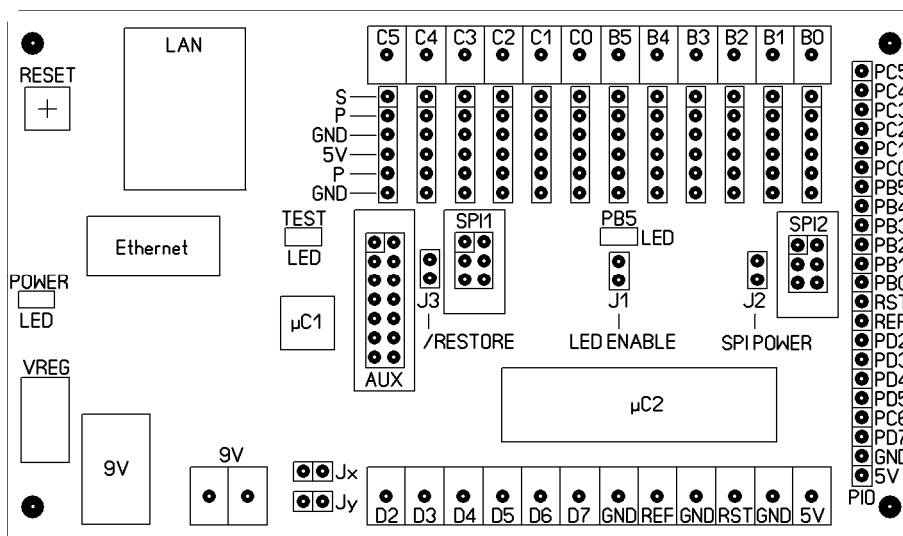
Das ABACOM netPIO-Bord ist ein 10-MBit-LAN-Netzwerkinterface mit der sich verschiedene Mess- und Steuerungsaufgaben durchführen lassen. Das Bord ist mit verschiedenen Firmware-Varianten erhältlich, die auch auf unseren USB- $\mu$ PIO-Modulen zum Einsatz kommen. Dieses Dokument beschreibt lediglich die Unterschiede und zusätzlichen Funktionen der Netzwerk-Variante. Informationen zur Verwendung der Port-Schraubanschlüsse finden Sie im Datenblatt zu Ihrer Firmware.

**Vor der ersten Inbetriebnahme sollten Sie die Anleitung einmal komplett gelesen haben.**

## Blockschaltbild (schematisch)



## Mechanischer Aufbau



Bordgröße 70mm x 120 mm (passt in WAGO-Hutschienenprofil WAGO 288-600)

### **Spannungsversorgung (Power)**

Das Modul benötigt eine externe Versorgungsspannung von 7-9V, die über eine übliche DC-Steckverbindung oder alternativ über einen Klemmenanschluss zugeführt werden kann.

Das Netzteil sollte wenigstens 500 mA liefern können. Die Ruhestromausnahme liegt bei typ. 200mA (ohne externe Beschaltung). Das Board besitzt eine eigene Spannungsstabilisierung (5V) mit vor geschaltetem Brückengleichrichter (Verpolungsschutz).

Die Verwendung stabilisierter DC-Netzteile ist zu empfehlen. Die Versorgungsspannung darf 9V nicht übersteigen, damit es nicht zu einer übermäßigen Erwärmung kommt. Externe Komponenten dürfen in der Summe das Board maximal mit 200mA belasten. Die Stromaufnahme pro Port darf 10mA nicht übersteigen. Sobald das Board versorgt wird leuchtet die grüne POWER-LED. Das sich die Platine - insbesondere der Spannungsregler (VREG) - erwärmt ist normal.

Wenn anderweitig eine externe, STABILISIERTE 5V-Spannungsversorgung zur Verfügung steht, kann das Board alternativ auch über die Anschlussklemmen GND/VCC(5V) versorgt werden. Ein Verpolungsschutz besteht hier nicht. Diese Methode ist vorteilhaft, denn sie vermeidet die Erwärmung, die sonst durch die Verlustleistung der Spannungsregelung entsteht.

### **Doppelcontroller**

Das netPIO verwendet zwei Mikrocontroller die nachfolgend als Kommunikationscontroller ( $\mu$ C1) und Anwendungscontroller ( $\mu$ C2) bezeichnet werden.

### **Netzwerkanschluss und Kommunikationscontroller**

Für den 10MBit-Netzwerkanschluss wird das Gerät mit Ihrem Router verbunden. Die grüne LED im LAN-Anschluss signalisiert den physikalisch korrekten Netzwerkanschluss. Eine gelbe LED im Anschluss signalisiert eingehende und ausgehende Datenpakete.

Als Interface-Chip dient ein MicroChip ENC28J70 Ethernet-Controller, der intern per SPI-Schnittstelle (SPI1) mit einem ATMEGA168-20 Microcontroller ( $\mu$ C1 - Communication Controller) verbunden ist. Dieser Kommunikationscontroller steuert die Netzwerkkommunikation. Für die Einstellung der Netzwerkparameter aller ABACOM netPIO Produkte steht eine gemeinsame einheitliche Windows-Konfigurationssoftware zur Verfügung. Der Kommunikationscontroller sendet und empfängt Netzwerk-Datenpakete und bereitet diese auf. Die Kommunikation erfolgt im wesentlichen über das UDP-Netzwerk-Protokoll.

Eine gelbe TEST-LED erleichtert die Geräte-Identifikation und Diagnose der Netzwerkeinstellungen in Verbindung mit der Konfigurationssoftware. Darüber hinaus stellt der Netzwerkkontroller einen digitalen IO-Port (AUX) mit jeweils sechs Ein- und Ausgängen zur Verfügung. Die genaue Funktion des AUX-Port wird später noch erläutert.

## **Anwendungscontroller (Application Controller)**

Ein zweiter Mikrocontroller ( $\mu C2$ ) - der sogenannte Anwendungscontroller - verfügt über drei I/O-Port B, C und D mit jeweils sechs I/O-Anschlüssen, über die das Modul die ihm zugeordnete Mess- und Steuerungsanwendung ausführt.

Die Datenkommunikation mit dem Kommunikationscontroller erfolgt intern über die seriellen Schnittstellenleitungen der beiden Chips mit 57600 Baud. Die interne serielle Verbindung ist über die Jumper Jx und Jy zugänglich. Als Anwendungscontroller kommt ebenfalls ein ATMEGA168-20 (hier in gesockelter Ausführung) zum Einsatz. Beide Controller verwenden eine Taktfrequenz von 18.432 MHz.

Die Verwendung von zwei Controllern ermöglicht es die aufwändige Netzwerkkommunikation vollständig von der (nicht minder aufwendigen) Steuerungsaufgabe weitestgehend zu trennen. Der Anwendungscontroller tauscht mit dem Kommunikationscontroller nur noch die relevanten Nutzdaten (intern per serieller Schnittstelle) aus und ist von der Datenaufbereitung der Netzwerkdatenpakete und der Netzwerkprotokolle völlig entlastet. Der Anwendungscontroller auf dem netPIO-Board verwendet exakt die gleiche Firmware, die auch auf unseren bewährten USB- $\mu$ PIO-Bords zum Einsatz kommt. Die Verwendung der Ports der Port ist ebenfalls somit ebenfalls identisch und der Beschreibung Ihrer Firmware zu entnehmen.

Folgende Firmwarevarianten sind erhältlich

/ **GPIO18** ([http://www.abacom-online.de/div/FirmwareDocs/DE/Firmware\\_GPIO18.pdf](http://www.abacom-online.de/div/FirmwareDocs/DE/Firmware_GPIO18.pdf))

### **Interface mit 18 programmierbaren I/O**

Sie verwenden das Board als I/O-Interface z.B. mit unserer ProfiLab-Software oder eigenen Programmen.

/ **PWMIO18** ([http://www.abacom-online.de/div/FirmwareDocs/DE/Firmware\\_PWMIO18.pdf](http://www.abacom-online.de/div/FirmwareDocs/DE/Firmware_PWMIO18.pdf))

### **Komfortables Interface mit 18 programmierbaren I/O**

mit erweiterten Funktionen, wie PWM, Servo, Blink-/ Pulsfunktionen.

Sie verwenden das Board als I/O-Interface z.B. mit unserer ProfiLab-Software oder eigenen Programmen.

/ **FREQ** ([http://www.abacom-online.de/div/FirmwareDocs/DE/Firmware\\_FREQ.pdf](http://www.abacom-online.de/div/FirmwareDocs/DE/Firmware_FREQ.pdf))

### **9-MHz.Frequenzzähler / Generator**

Sie verwenden das Board als Frequenzzähler z.B. mit unserer ProfiLab-Software oder eigenen Programmen.

/ **INCR3** ([http://www.abacom-online.de/div/FirmwareDocs/DE/Firmware\\_INCR3.pdf](http://www.abacom-online.de/div/FirmwareDocs/DE/Firmware_INCR3.pdf))

### **Drei Inkrementalgeber-Eingänge**

Sie verwenden das Board mit bis zu drei Inkrementalgebern z.B. mit unserer ProfiLab-Software oder eigenen Programmen.

/ **TEMP12** ([http://www.abacom-online.de/div/FirmwareDocs/DE/Firmware\\_TEMP12.pdf](http://www.abacom-online.de/div/FirmwareDocs/DE/Firmware_TEMP12.pdf))

Temperatursystem für bis zu 12x Dallas DS18B20 Temperatur-Sensoren

Die Schnittstelle SPI2 des Anwendungscontrollers dient (ebenso wie die SPI1 des Kommunikationscontrollers) zur hauseigenen Programmierung der Controller-Chip-Firmware. Eine Verwendungsmöglichkeit für den Endanwender besteht nicht.

## **I/O-Ports - Die Verbindung zu Aussenwelt**

Folgende I/O-Leitungen des Anwendungscontrollers stehen zur Verfügung:

- Port B: Signalleitung B0...B5
- Port C: Signalleitung C0...C5
- Port D: Signalleitung D2...D7

Über die Funktion der Signalleitungen entscheidet die Firmware des Anwendungscontrollers mit der Sie das netPIO-Bord bezogen haben. Alle Portleitungen arbeiten mit TTL-Pegeln, d.h. die Spannungen an den Portleitungen dürfen die Betriebsspannung (+5V) nicht übersteigen und das Massepotential (0V; GND) keinesfalls unterschreiten.

Alle genannten Portleitungen stehen direkt an der PIO-Anschluss-Reihe zur Verfügung, die es dem Anwender ermöglicht wahlweise Stift- oder Buchsenleisten, Drähte oder andere Komponenten einzulöten. Port D steht zusätzlich DIREKT an den entsprechenden Schraubklemmen zu Verfügung. Port B und Port C verfügen darüber hinaus jeweils noch über eine Stiftleistenreihe (Jumper Field).

Die Verwendung der Stiftleiten erleichtert in vielen Fällen die externe Beschaltung. Die Stiftleisten sind beginnend an der Schraubklemme in dieser Reihenfolge belegt:

- 1.) **S =Schraube (Screw)** ist mit der zugehörigen Schraubklemme verbunden.
- 2.) **P - Port** ist direkt mit der zugehörigen I/O-Portleitung verbunden.
- 3.) **GND - Masse (Ground)** ist ein Masseanschluss.
- 4.) **+5V** ist mit der 5V-Versorgungsspannung des Bords verbunden.
- 5.) **P - Port** ist wiederum die zugehörige I/O-Portleitung des Anwendungscontrollers.
- 6.) **GND - Masse (Ground)** Ist wiederum ein Masseanschluss.

Achten Sie beim Anschluss externer Komponenten immer darauf, das Sie die Belastungsgrenzen einhalten:

- max. 10 mA pro Port
  - in der Summe nicht mehr als 200mA externe Last
- Externe Beschaltungen mit grösserem Strombedarf müssen extern selbst versorgt werden!

Die Anwendungen für die Stiftleisten sind vielfältig:

Beispiele 1: Jumper von S nach P

S und P sind mit einem Jumper verbunden (Werkseinstellung). Die Schraubklemme ist nun mit der Portleitung verbunden.

Beispiel 2: Jumper von P nach GND

Eine Eingangs(!)leitung wird definiert auf Masse-Potential (low) gelegt.

Beispiel 3: Jumper von +5V nach P

Eine Eingangs(!)leitung wird definiert auf Plus-Potential (high) gelegt.

Beispiel 4: Ein steckbarer Sensor mit der Anschlussreihenfolge 5V - Data - GND kann einfach auf die Pins 4,5 und 6 der Stiftleite gesteckt werden.

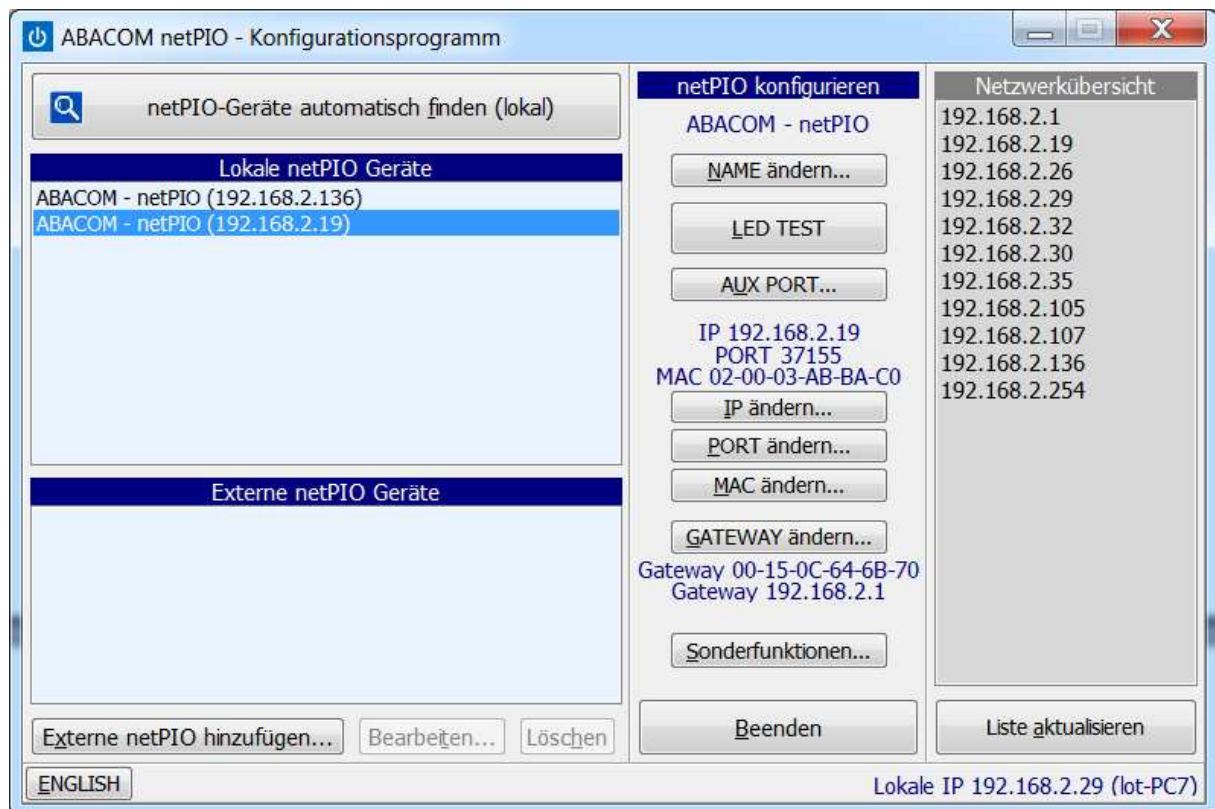
Beispiel 5: Ein eigene Eingangsbeschaltung (z.B. Überspannungsschutz, Signalverstärker, Abschwächer, o.ä.) kann zwischen Schraubklemme (Pin 1) und Port (Pin 2) eingeschleift werden, ohne den Schraubanschluss lösen zu müssen.

Beispiel 6: Eine eigene, selbstentworfene Huckepackplatine könnte aufgesteckt werden.

## Netzwerkconfiguration

Verbinden Sie den LAN-Anschluss mit Ihrem Router und versorgen Sie das Bord mit der Betriebsspannung. Der Router muss die Datenrate 10MBit unterstützen, was bei fast allen marktgängigen Geräten der Fall ist. Die POWER-LED (grün) und die CONNECTION-LED (grün) am LAN-Anschluss leuchten auf, wenn der Anschluss korrekt erfolgt ist. Die gelbe LED am LAN-Anschluss zeigt sofern vorhanden Datenpakete vom Netzwerk an.

Schliessen Sie zunächst immer nur EIN neues netPIO Gerät an Ihr Netzwerk. Weitere dürfen erst folgen wenn die Netzwerkkonfiguration für das neue Gerät erfolgreich abgeschlossen wurde.



Die Netzwerkkonfiguration erfolgt für alle ABACOM netPIO-Produkte in gleicher Weise, mit Hilfe des Konfigurationsprogramms (netPIO\_config.exe). Das Konfigurationsprogramm muss auf einem Windows-PC ausgeführt werden, welcher sich im gleichen Netzwerk wie das neue netPIO-Gerät befindet.

Das Konfigurationsprogramm erkennt und verwaltet die Netzwerkkonfiguration aller im lokalen Netzwerk vorhandenen und neuen netPIO-Geräte. Die Erkennung erfolgt automatisch bei Programmstart oder einen Klick auf die entsprechende Schaltfläche.

Neue Geräte die sich im Auslieferungszustand befinden werden mit einer freien IP in Ihrem Netzwerk angemeldet. Das neue Geräte erscheint daraufhin in der Liste der lokalen netPIO-Geräte. Die verwendete Standard-Portadresse ist 37155 (UDP-Port).

Das Konfigurationsprogramm erfüllt folgende Aufgaben:

- Einstellung der Netzwerkparameter ( IP, Port, Gateway, ggf. Ändern der MAC-Adresse )

- Vergabe benutzerfreundlicher Gerätenamen
- Status-Anzeige des AUX-Ports
- Master-Slave-Konfiguration (Traps)
- Aufruf weiterer Sonderfunktionen

Mit Ausnahme der Netzwerkparameter können auch Einstellungen an entfernt stehenden Geräten z.B. per Internet vorgenommen werden, wenn für diese ein entsprechender Zugang zuvor eingerichtet wurde.

Sofern erforderlich oder gewünscht können Sie nun die IP-Adresse und die Portnummer nach Bedarf ändern. Markieren Sie dazu das Gerät in der Liste der lokalen netPIO-Geräte, und betätigen Sie dann die gewünschte Konfigurationsschaltfläche (z.B. "IP ändern").

Die Schaltfläche LED TEST schaltet die TEST-LED auf dem Modul ein und aus. So können Sie leicht die Netzwerkfunktion und Zuordnung überprüfen.

netPIO-Geräte verwenden eine feste IP-Adresse (kein DHCP). Um Sie bei der Vergabe einer neuen IP zu unterstützen, zeigt das Konfigurationsprogramm eine Liste IP-Geräten die z.Zt. im Netzwerk aktiv sind.

Sobald Sie ein neues Gerät erfolgreich mit einer IP im Netzwerk angemeldet haben, können Sie ggf. nun weitere Geräte - eins nach dem andern - anschliessen und wie zuvor beschrieben konfigurieren.

Die Vergabe eines Gerätenamens dient der Benutzerfreundlichkeit und hat keinen Einfluss auf die Funktionalität der Geräte. Die Schaltfläche "NAMEN ändern" erlaubt die Vergabe eigener Gerätenamen, wie z.B. "Büro #1", "Garage", "Heizung", etc.

MAC-Adresse und Port müssen nur in (seltenen) Fällen verändert werden, wenn Netzwerkkonflikte bestehen.

Die Gateway-Adresse ist die lokale IP-Adresse Ihres Routers (z.B. 192.168.1.1). Diese Adresse kann bei Bedarf umkonfiguriert werden.

Alle Konfigurationseinstellungen werden im EEPROM-Speicher des Kommunikationskontrollers abgelegt und dauerhaft gespeichert. Sollte es in Ausnahmefällen passieren das ein Gerät nicht mehr reagiert oder gefunden wird, betätigen Sie den RESET-Taster am Gerät. Danach sollte das Gerät von der Konfigurationssoftware wieder erkannt werden. Die Konfigurationsdaten bleiben dabei erhalten.

In Fällen in denen diese Vorgehensweise nicht zum Erfolg führt und andere Fehlerquellen auszuschliessen, wurde vermutlich eine falsche Netzwerkkonfiguration vorgenommen. In diesem Fall muss das Gerät in den Auslieferungszustand zurück gesetzt werden und die Konfiguration wie bei einem Neugerät erneut vorgenommen werden.

Um den Auslieferungszustand wieder herzustellen...

- entfernen Sie den Jumper J3 (/Restore; zwischen AUX-Port und SPI1)
- Betätigen Sie nun die RESET-Taste am Gerät (Spannungsversorgung muss vorhanden sein)
- anschliessend stecken Sie den Jumper J3 wieder auf
- abschliessend betätigen nochmals den RESET-Taster

Das Gerät befindet sich nun wieder im Auslieferungszustand. Führen Sie nun die Netzwerk-Konfiguration wie zuvor für ein Neugerät beschrieben erneut durch.

#### Liste externer netPIO-Geräte

In dieser Liste können netPIO-Geräte manuell mit Hostnamen (oder IP) und Port angemeldet werden, die sich ausserhalb Ihres lokalen Netzwerks befinden, also z.B. Geräte für die Sie einen Fernzugriff per Internet konfiguriert haben. Die Netzwerkkonfiguration dieser Geräte muss zuvor lokal vorgenommen werden.

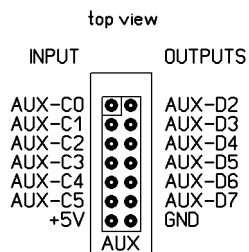
Für externe Geräte sind einige Funktionen der Konfigurationssoftware gesperrt, um z.B. das Verändern der IP-Adresse oder der Porteinstellung per Fernzugriff zu verhindern. (Nach einer solchen Änderung wäre das Gerät sonst nicht mehr per Fernzugriff erreichbar.)



## AUX-Port

Am Anschluss AUX aller netPIO-Geräte stehen Ihnen zusätzlich sechs digitale Eingänge und sechs digitale Ausgänge zur Verfügung. Diese sind unmittelbar mit dem (fest verbauten) Kommunikationskontroller verbunden. Ausserdem steht hier die 5V-Betriebsspannung und ein Masseanschluss zur Verfügung. Bei der Verwendung dieser Anschlüsse ist daher besondere Sorgfalt geboten. Fehlerhafte Beschaltungen, sowie Über- und Unterspannungen können den Controller beschädigen, der nicht ausgetauscht werden kann.

Alle Portleitungen arbeiten auch hier mit TTL-Pegeln, d.h. die Spannungen an den Portleitungen dürfen die Betriebsspannung (+5V) nicht übersteigen und das Massepotential (0V; GND) keinesfalls unterschreiten. Bitte verwechseln Sie die AUX-Kanäle nicht mit den Ports des Anwendungscontrollers ( z.B. "AUX-D2" ist NICHT GLEICH "D2")



Die Pin-Belegung des AUX-Port

- Ansicht von der Bauteilseite
- Links die Eingänge
- Rechts die Ausgänge

Offene Eingänge des AUX-Port sind intern über Pullup-Widerstände auf HIGH-Potential gelegt, und können extern z.B. über Taster oder Open-Collector-Ausgänge bedient (d.h. nach Masse gezogen) werden. Ausgänge des AUX-Port können in der Konfigurationssoftware ein-, aus- und umgeschaltet werden. Die Signalzustände der Ein- und Ausgänge werden in der Konfigurationssoftware angezeigt.

Somit bietet der AUX-Port auf den ersten Blick lediglich weitere (Hilfs-) I/O-Leitungen an, die unabhängig von der Firmware des Anwendungscontrollers zur Verfügung stehen.

Doch der AUX-Port hält noch ein weiteres Leistungsmerkmal der netPIO bereit...

### **Signalgesteuerte Botschaften (Traps)**

Der Netzwerkkontroller erkennt Änderungen am Signalzustand der AUX-Signale und der TEST-LED und sendet ggf. per Netzwerk eine Benachrichtigungsbotschaft an ein anderes netPIO-Gerät oder auch sich selbst. Der Empfänger dieser Botschaft führt daraufhin einen Schaltvorgang an einem seiner Signalausgänge oder der TEST-LED aus.

Das Gerät welches die Nachricht aussendet wird als MASTER bezeichnet. Das Gerät das die Botschaft empfängt und ausführt wird als SLAVE bezeichnet. Derartige Benachrichtigungsbotschaften, die ohne vorherige Anforderung ins Netzwerk geschickt werden, bezeichnet man als Traps (Fallen).

Jedes netPIO-Gerät erlaubt es maximal sechs verschiedene Traps (Botschaften) zu definieren und zu speichern.

Das Senden einer Trap (durch den Master) wird durch einen Zustandswechsel (steigende UND fallende Flanke) eines Triggersignals ausgelöst. Als Triggersignal können alle Signale des AUX-Ports und die TEST-LED verwendet werden. D.h. eine Triggerung kann z.B. durch ein Eingangssignal (z.B. einen Taster) ausgelöst werden. Ebenso kann aber auch das Umschalten eines Ausgangssignal eine Nachricht auslösen.

Die Trap-Nachricht die versendet wird ist immer an einen bestimmten Empfänger (Slave) gerichtet. Der Slave der die Nachricht erhalten soll (also irgendein netPIO-Gerät das per Netzwerk erreichbar ist) wird durch seine Netzwerkadresse (IP oder Hostname und Portadresse) adressiert. Die Adressierung lokal direkt über die Geräte-IP erfolgen. Es können aber auch Geräte per Hostnamen (z.B. per DynDNS-Hostnamen) angesprochen werden.

Die Nachricht enthält ausserdem noch die Anweisung für den Slave, die dieser bei Erhalte der Nachricht ausführen soll. Mögliche Anweisungen an den Slave sind das EIN-, AUS- oder Umschalten eines seiner AUX-Ausgänge und die Anweisung dem Zustand des Triggersignals zu FOLGEN.

#### Beispiel 1:

Gerät #1 erkennt den Zustandswechsel am AUX-Eingang C0 (z.B. ein Taster).

Gerät #1 sendet eine Nachricht an Gerät #2 mit der Anweisung "Schalte deinen AUX-AUSGANG D0 UM!"

Gerät #2 empfängt die Anweisung und führt diese aus (z.B. Umschalten eines Relais).

#### Beispiel 2:

Gerät #1 erkennt das Einschalten seiner TEST-LED.

Gerät #1 sendet eine Nachricht an Gerät #2 mit der Anweisung "FOLGE MIR MIT DEINER TEST-LED!"

Gerät #2 empfängt die Anweisung und schaltet seinerseits seine TEST-LED ein.

Dem aufmerksamen Leser werden die umfangreichen Möglichkeiten dieser Technik nicht entgangen sein, welche es ermöglicht Signale direkt von einem netPIO-Modul auf eine anderes zu übermitteln, ohne das dazu ein PC als Vermittler bemüht werden muss.

Allerdings besteht auch die mögliche Gefahr, dass durch fehlerhafte Konfiguration Nachrichten-Schleifen (Rückkopplungen) entstehen, die dann dazu führen das das Netzwerk mit Nachrichten überflutet wird, die sich gegenseitig immer wieder neu auslösen. Planen Sie Ihre Traps also sorgfältig!

Ein Beispiel was man BESSER NICHT machen sollte:

Gerät #1 erkennt den Zustandswechsel seiner TEST-LED

Gerät #1 sendet an Gerät #2 die Nachricht "SCHALTE DEINE TEST-LED UM!"

Gerät #2 erkennt den Zustandswechsel seiner TEST-LED

Gerät #2 sendet an Gerät #1 die Nachricht "SCHALTE DEINE TEST-LED UM!"

Gerät #1 erkennt den Zustandswechsel seiner TEST-LED

usw.

Einmal angestossen benachrichtigen sich hier zwei Geräte immer aufs Neue, bis der Netzwerkstecker gezogen wird!

### **Sonderfunktionen**

Abschliessend ermöglicht die Konfigurationssoftware noch eine der folgenden Sonderfunktionen durchzuführen:

- RESET des Moduls per Software (entspricht dem Drücken der RESET-Taste am Gerät)
- RESET des Anwendungskontrollers
- WERKSEINSTELLUNG WIEDERHERSTELLEN, löscht alle Konfigurationsdaten aus dem EEPROM des Gerätes und versetzt das Gerät zurück in den Auslieferungszustand.

### **Firmware der Anwendungscontrollern**

Das Board wird fertig programmiert in verschiedenen Firmware-Varianten angeboten, welche auch auf unseren USB-Modulen (ABACOM USB- $\mu$ PIO) zum Einsatz kommt.

Eine detaillierte Beschreibung der Firmware und die dazugehörige Funktion und die Belegung der I/O-Port finden Sie daher im Datenblatt der USB-Module.

Die Firmware und die Ports der netPIO und der USB- $\mu$ PIO sind **identisch**.

## **Programmierung**

### **Kommunikation mit dem Anwendungscontroller**

Bei der Programmierung kommen die gleichen Protokolle zum Einsatz, die im Datenblatt der USB- $\mu$ PIO beschrieben sind.

Die netPIO **kapselt** die seriellen Protokolle der USB-Variante, und überträgt diese per UDP-Netzwerkprotokoll (User Datagram Protokoll). UDP ist ein einfaches (verbindungsloses) Netzwerkprotokoll mit dem kleine Datenpakete (Datagramme) zwischen Netzwerkteilnehmern ausgetauscht werden können.

Das Prinzip der Frage-Antwort-Übertragung (Request-Response) der USB-Module bleibt dabei erhalten.

Programme die für ein netPIO entwickelt werden sollen, müssen die also die Requests in einem UDP-Datenpaket ablegen und übertragen. Das Gerät sendet daraufhin ein UDP-Antwortpaket zurück, welches die Response-Daten enthält.

### **Request per UDP**

Bei der Kapselung eines Requests (der USB-Variante) wird dieser mit einigen, wenigen Zusatzdaten im Datagramm des UDP-Pakets abgelegt:

- Dem ursprünglichen Request wird die ASCII-Zeichenfolge "TXDATA" als Steuerkommando vorangestellt, welche dem Kommunikationskontroller mitteilt, den eigentlichen Request (intern per UART) an den Anwendungscontroller weiterzureichen.
- Zusätzlich werden noch zwei weitere Bytes an das Datenpaket angehängt. Ein Byte für die Länge der erwarteten Response und ein SYNC-Byte.

Das Datagramm das im UDP-Datenpaket übertragen werden muss ist also immer um acht Bytes länger als der ursprüngliche, gekapselte Request, nämlich 6 Bytes für die Zeichenfolge "TXDATA" und zwei Bytes für die Antwortlänge und das SYNC-Byte.

`<"TXDATA"(6 Bytes)> + <Request(n Bytes)> + <Antwortlänge(1 Byte)> + <Sync(1 Byte) >`

### **Response per UDP**

Nachdem das UDP-Paket verschickt wurde antwortet der Controller mit einem UDP-Datenpaket, welches die Response enthält. Das Antwort-Datagramm hat den gleichen Aufbau wie die Response der USB-Variante. Bei der Antwort ist keine Kapselung notwendig.

### **Hinweise zur Netzwerkprogrammierung**

Zum Versenden muss das UDP-Datenpaket noch mit der Empfänger-IP und dem verwendeten UDP-Port adressiert werden. Je nach Programmiersprache wird man dabei auf fertige UDP-Komponenten zurückgreifen, die den Versand und Empfang der UDP-Pakete übernehmen. Die bekanntesten sind vermutlich die Indy-Netwerkkomponenten.

```
...  
IndyUDP.Host = "192.168.1.222"  
IndyUDP.Port = 37155  
IndyUDP.SendBuffer(RequestDatagramm)  
...  
IndyUDP.ReceiveBuffer(ResponseDatagramm)  
...
```

Bitte haben Sie Verständnis, dass wir keine Fragen zu UDP-Komponenten (wie. z.B. Indy) oder zu den vielfältigen Programmiersprachen, Versionen und Dialekten beantworten können.

Fragen wie z.B. "Wie verschicke ich ein UDP-Paket mit Visual Basic 6.0?" oder "Wie baue ich ein Request-Datenpaket in Java?" werden auch wir Ihnen nicht beantworten können. Hobby-Programmierer finden Antworten auf derartige Fragen bitte in den einschlägigen Foren und Tutorials, die es zu jeder Programmiersprache zahlreich im Internet gibt. Programmierbeispiele stehen aus o.g. Gründen ebenfalls nicht zur Verfügung.

## Weitere Funktionen des Kommunikationscontrollers

Der Kommunikationscontroller kann weitere Steuerkommandos verarbeiten, die ebenfalls per UDP-Datagramm verschickt werden. Das Kommando "TXDATA" für die Datenübertragung zum Anwendungscontroller haben zuvor bereits kennengelernt.

Weitere Steuerkommandos für den Kommunikationscontroller sind:

### "LEDTOGGLE"

Schaltet die TEST-LED um (ein/aus).

### "LEDON"

Schaltet die TEST-LED ein.

### "LEDOFF"

Schaltet die TEST-LED aus.

### "netPIO?"

Gerät antwortet mit "netPIO!"

### "SLFRST"

Das netPIO führt einen Neustart durch (RESET).

### "RSTCTRL"

Der Anwendungscontroller führt einen Neustart aus.

### "RSTFAC"

Löscht alle Benutzereinstellungen im Gerät und stellt die Werkseinstellung wieder her.

### "CHANGEIP" + 4 Bytes mit der neuen IP-Adresse

Ändert die IP-Adresse des Gerätes.

### "CHANGEPORT" + 2 Bytes mit der neuen Port-Nummer

Ändert den UDP-Port des Gerätes.

### "CHANGEMAC" + 6 Bytes mit der neuen MAC-Adresse

Ändert die MAC-Adresse des Gerätes.

### "GATEWAY" + 4 Bytes mit der neuen Gateway-IP

Ändert die Gateway-Adresse (=Router-IP) für den Internetzugang.

### "AUXD<sub>x</sub>ON"

Schaltet einen AUX-Port-Ausgang auf high. ("x" = "2" ... "7")

### "AUXD<sub>x</sub>OFF"

Schaltet einen AUX-Port-Ausgang auf low. ("x" = "2" ... "7")

### "AUXD<sub>x</sub>TOGGLE"

Schaltet einen AUX-Port-Ausgang auf um. ("x" = "2" ... "7")

## "GETAUX"

Fragt den Status des AUX-Ports und der TEST-LED ab. Die Antwort ist 4 Bytes lang. Die vier Bytes bilden einen lesbaren ASCII-String (z.B. "F3FC" ). Der Wert des oberen Bytes (hier "F3") repräsentiert den Eingangsstatus, der Wert des unteren Bytes (hier "FC") den Ausgangsstatus des AUX-Ports:

Low-Byte : AUX-C(Bit 0..5) Eingangsstatus

High-Byte: AUX-D(Bit 2..7) Ausgangsstatus und TEST-LED (Bit 0)

**"TXDATA" + n-Bytes Request + 1 Byte ResponseLen + 1 Byte Sync**

Überträgt den Request (n-Bytes) intern per UART (57600 Baud) an den

Anwendungscontroller (d.h. ohne "TXDATA", ohne ResponseLen und ohne SYNC-Byte).

ResponseLen<>0 (= 1 ... 254):

Erwartet eine Antwort vom Anwendungscontroller mit der Länge ResponseLen.

Sobald die Antwort vom Anwendungscontroller vorliegt wird diese unverändert als UDP Antwort zurückgesendet.

ResponseLen=255:

Verwendet den Wert des SYNC-Bytes als Daten-Ende-Zeichen. Nimmt eine Antwort vom Anwendungscontroller entgegen, bis dieser abschliessend das SYNC-Byte sendet. Sobald die Antwort vom Anwendungscontroller vorliegt wird diese unverändert als UDP-Antwort zurückgesendet.

ResponseLen=0 ist nicht erlaubt.