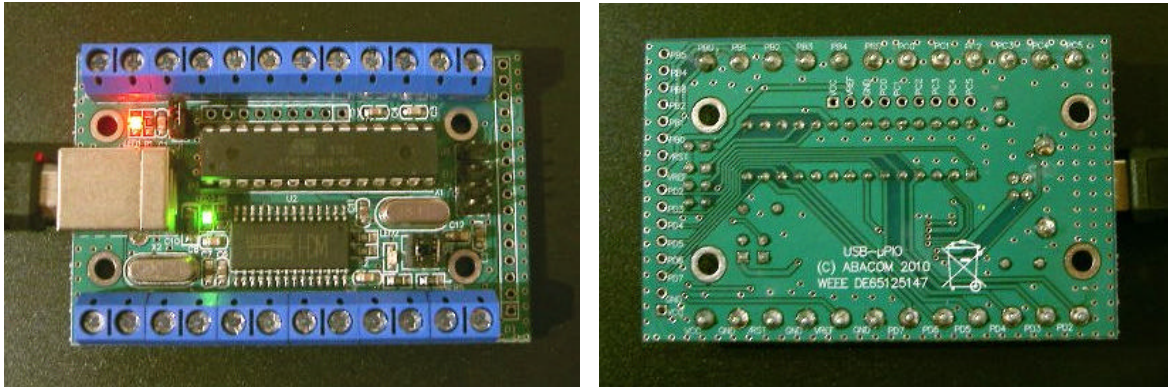


# USB- $\mu$ PIO



## Das USB-AVR-Board für Praktiker

- Kompaktes AVR-Board mit Atmel ATmega168-20
- High Speed Taktfrequenz 18.432000 MHz
- 100% error free High Baudrates
- Komfortabler Anschluss über Schraubklemmen oder Stiftleisten
- 6-poliger ISP-Anschluss
- Betriebsspannung (+5V) wahlweise vom USB oder extern
- 3 LED: USB-Power, COM (RxD) , USER (Port B.5; deaktivierbar)
- SOFT-RESET per DTR-Signal (deaktivierbar)
- Standard USB-B-Anschluss
- USB Chipsatz: CH341A
- Virtueller (RS232-)-COM-Treiber (VCP)
- Systemvoraussetzungen: XP, 2000, Vista, WIN7, 32/64 Bit
- Abmessungen ca. 70 x 45 x 15 mm

## Firmware-Optionen

Das Board wird von uns - fertig programmiert - mit verschiedenen Firmwarevarianten angeboten:

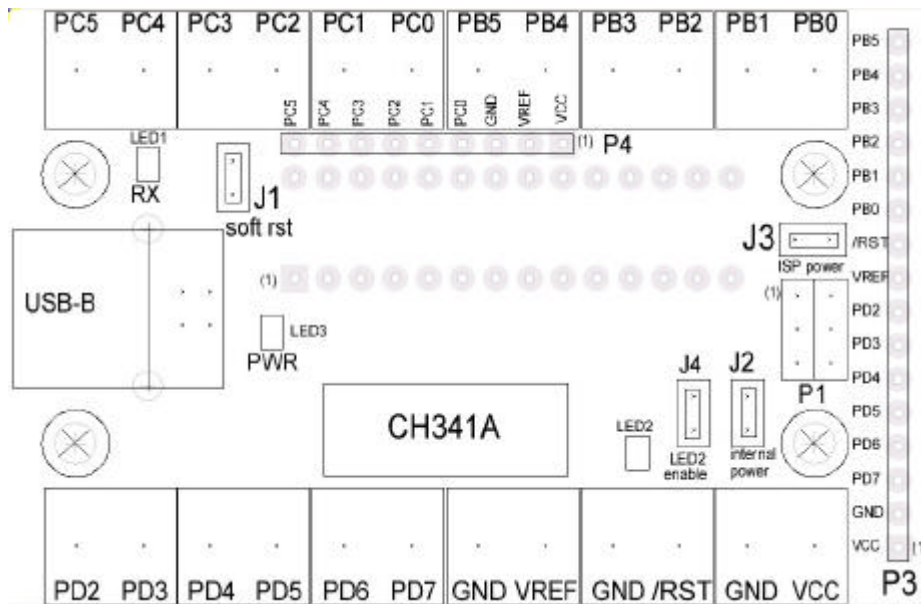
/GPIO18	I/O –Interface mit 18 programmierbaren I/O
/FREQ	9-MHz.Frequenzzähler / Generator
/INCR3	3 Inkrementalgeber-Eingänge
/TEMP12	Temperatursystem für Dallas DS18B20 Sensoren
/ABADUINO	BOOTLOADER für Arduino-Software ( <a href="http://www.arduino.cc/">http://www.arduino.cc/</a> ) Das Board ähnelt in Teilen den Arduino USB Schaltplänen.
/MCS	MCS BOOTLOADER (BASCOM AVR)
/CLEAR	Nur CHIP - unprogrammiert *

\* für die Programmierung ist ein ISP-Programmer erforderlich

## Download

[http://www.abacom-online.de/div/setup\\_usb\\_μPIO.exe](http://www.abacom-online.de/div/setup_usb_μPIO.exe)

## Anschlüsse, LED und Jumper



P1 – ISP-Anschluss für Programmierer

P2 – USB-B-Anschluss

P3 – Anschlussleiste Port B, Port D, VREF, RST, VCC, GND (Digitalport)

P4 – Anschlussleiste Port C, VREF, VCC, GND (Digitalport und ADC-Analogport)

J1 – Jumper SOFT RST verbindet RESET(PC6) mit DTR-Signal vom CH341A

J2 – Jumper INTERNAL POWER verbindet VCC mit USB-Versorgungsspannung

J3 – Jumper ISP POWER verbindet VCC mit ISP

J4 – Jumper LED2 ENABLE verbindet LED2 mit PB5

LED1 – RX Indikator (rot; Datenempfang vom PC)

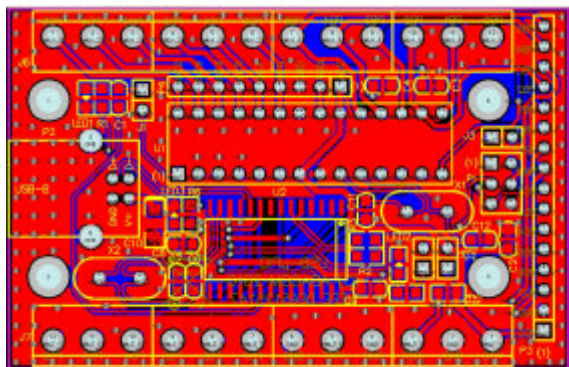
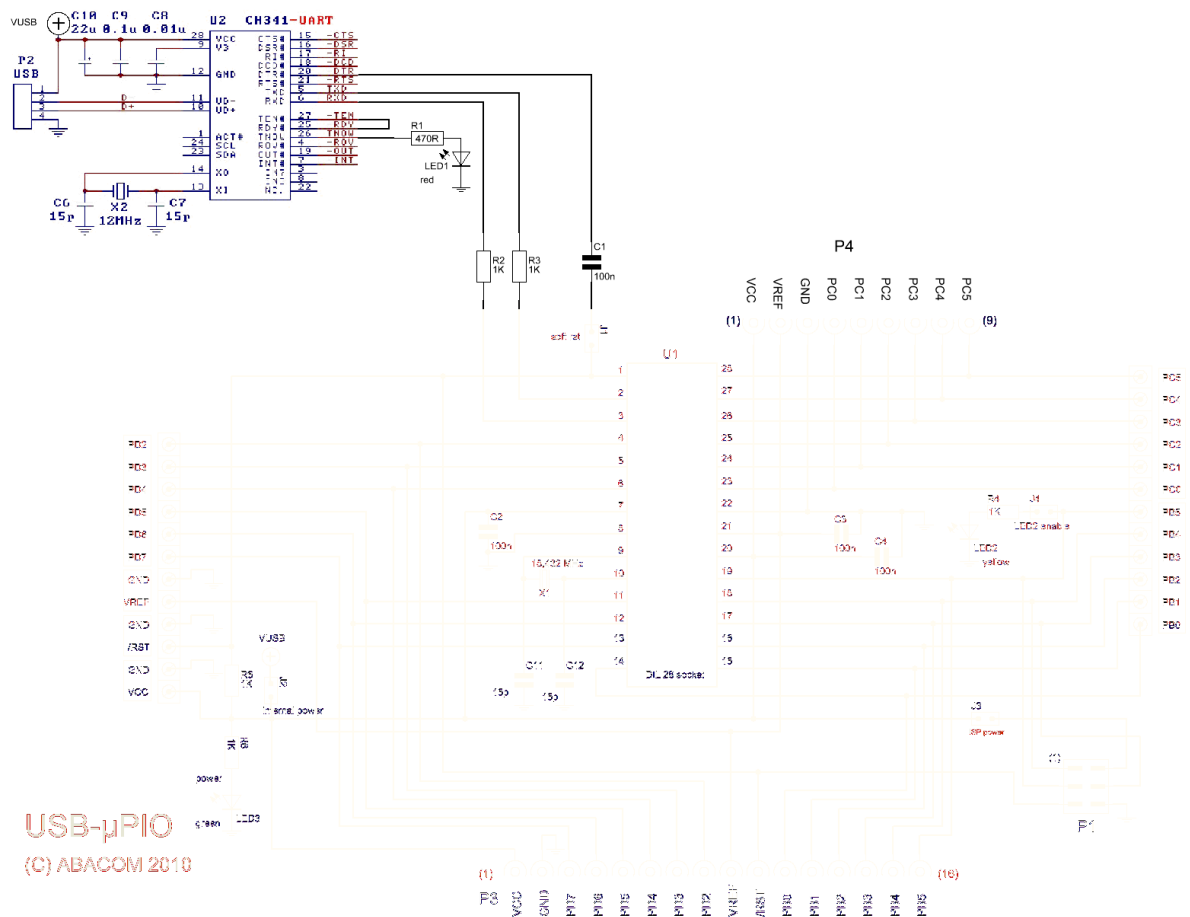
LED2 – LED an PB5 (gelb; für eigene Zwecke)

LED3 – USB-Power (grün; USB-Spannungsversorgung)

Schraubklemmen für Port B, Port C und Port D

### Die Anschlussbezeichnungen entsprechen denen des ATmega168

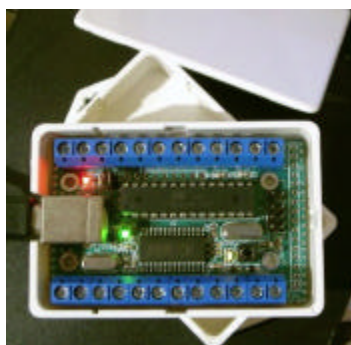
(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)



Abmessungen ca. 70 x 45 x 15 mm

### Als Zubehör erhältlich:

- Modulgehäuse (unbearbeitet)
- USB-Programmer inkl. ISP-Kabel



## Firmware

Das Board wird von uns - fertig programmiert - mit verschiedenen Firmwarevarianten angeboten:

### / GPIO18

#### I/O INTERFACE

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

### / FREQ

#### FRQUENZZÄHLER / GENERATOR

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

### / INCR

#### INKREMENTALGEBER

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

### /TEMP12

#### TEMPERATURSYSTEM

für bis zu zwölf Dallas DS18B20 Sensoren

### /ABADUINO

BOOTLOADER für Arduino-Software (<http://www.arduino.cc>)

Das Board ähnelt in Teilen den Arduino USB Schaltplänen. Der Bootloader wurde für die geänderte Taktfrequenz von 18.342 MHz modifiziert. Die Bootloader-Source ist im Software-Download enthalten und öffentlich gemacht. Sie finden diese zusammen mit der Installationsanleitung im Ordner SOFTWARE\ABADUINO.

### /MCS

#### MCS BOOTLOADER (BASCOSM AVR)

Sie verwenden BASCOSM-AVR für die Programmierung eigener Anwendungen. Durch die Verwendung des MCS-Bootloaders benötigen KEINEN ISP-Programmer. Ihre Anwendung wird einfach per USB direkt auf den Chip übertragen.

### /CLEAR

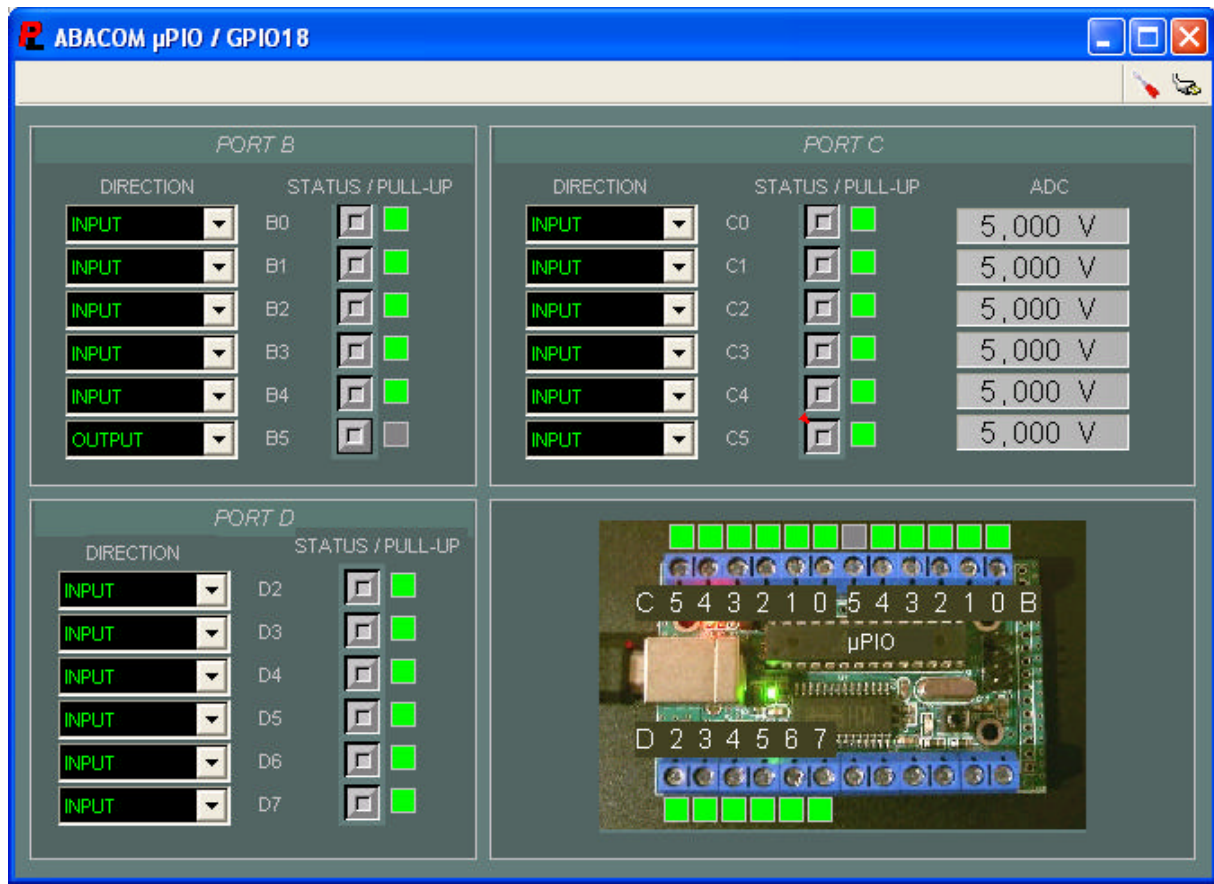
Nur CHIP - unprogrammiert \*

\* für die Programmierung ist ein ISP-Programmer erforderlich

Sie haben Erfahrung mit der Programmierung des Atmel ATmega168 und programmieren / flashen Ihre Firmware selbst.

Im Anschluss erfolgt eine ausführliche Beschreibung der Firmwarevarianten. Abhängig davon welche Variante Sie erworben haben, ist immer nur eines der nachstehenden Kapitel für Sie zutreffend.

## μPIO / GPIO18



Das μPIO / GPIO18 arbeitet als PC-Interface mit 18 digitalen I/O-Kanälen deren Datenrichtung umschaltbar ist (18 GPIO).

Die I/O-Leitungen sind zu drei Ports zusammengefasst (Port B, Port C, Port D). Jeder Port besitzt sechs I/O-Leitungen:

Port B0...PortB5	Digital-I/O
Port C0...PortC5	Digital-I/O = ADC 0..5
Port D2...PortD7	Digital-I/O

Die sechs Spannungen am Port C können darüber hinaus mit dem internen 10-Bit-A/D-Wandler (ADC) eingelesen werden. Die Referenzspannung für den ADC kann dabei aus drei umschaltbaren Quellen bezogen werden:

Externe Referenzspannung:

Die Referenzspannung wird über den Anschluss VREF eingespeist.  
 $0 < V_{ref} < V_{cc}$  ist dabei einzuhalten.

Interne Referenz VCC

Als Referenzspannung dient die positiver Versorgungsspannung (also ca. 5V).

Interne Referenz 1.1 V

Eine Referenzspannung von 1.1V wird intern erzeugt und dient als Referenz.

Für Kanäle die als Eingang arbeiten kann per Software ein interner Pullup-Widerstand aktiviert werden, der den jeweiligen Eingang auf das Potential der positiven Versorgungsspannung ( $V_{cc}$ ) legt, um z.B. offenen Eingänge ein definiertes Potential zu geben.

Eine kostenlose PC-Software steht zur Verfügung. Diese ermöglicht neben der Bedienung auch die einfache Datenabfrage per DDE aus anderen Programmen.

## Datenprotokoll

Die Kommunikation mit dem PC erfolgt per USB. Der USB-Treiber installiert einen virtuellen COM-Port. Der Datenaustausch erfolgt somit wie über eine ‚echte‘ RS232-Schnittstelle.

### Die Schnittstellenparameter sind: 57600, 8 N, 1

Baud 57600  
8 Datenbits  
Keine Parität  
1 Stopbit

Die Kommunikation erfolgt nach dem Frage-Antwort-Prinzip (Request/Response). Dabei besteht ein Request immer aus genau sieben Datenbytes. Nachdem Empfang dieser sieben Datenbytes sendet die Firmware 15 Bytes als Antwort an den PC zurück. Die erzielbare Abfragerate (sample rate) liegt bei etwa 150 Samples/Sek.

### Request: 7 Bytes

Byte1: ADC-Referenzumschaltung (0=EXTERN; 1=INTERN VCC; 3=INTERN 1.1V)  
Byte2: Richtung Port B  
Byte3: Richtung Port C  
Byte4: Richtung Port D  
Byte5: Daten Port B  
Byte6: Daten Port C  
Byte7: Daten Port D

Die Bytes 2..4 geben die **Datenrichtung** an. Ein gesetztes Bit setzt die zugehörige Portleitung auf AUSGANG, ein gelöschtes Bit setzt die Portleitung auf EINGANG.

Die Bytes 5..7 setzen den Ausgangszustand der I/O-Leitungen deren Datenrichtung auf AUSGANG gesetzt ist. Ein gesetztes Bit schalten den Ausgang HIGH (5V), ein gelöschtes Bit schaltet den Ausgang LOW (0V). Für I/O-Leitungen die nicht als Ausgang sondern als EINGANG definiert wurden, wird mit den Datenbytes (Byte 5...Byte7) ein interner Pullup-Widerstand aktiviert. Ein gesetztes Bit aktiviert den Pullup für eine I/O-Leitung. Ein gelöschtes Bit deaktiviert den Pullup für die zugehörige I/O-Leitung.

Die Bits-Nummern der Datenbytes sind den Leitungsnummern der Portleitungen zugeordnet. Ein gesetztes Bit2 in Byte4 eines Requests setzt also z.B. die Datenrichtung der Portleitung D2 auf Ausgang.

### Response: 15 Bytes

Byte 1: Status Port B  
Byte 2: Status Port C  
Byte 3: Status Port D  
Byte 4: ADC0 LSB  
Byte 5: ADC0 MSB  
Byte 6: ADC1 LSB  
Byte 7: ADC1 MSB  
Byte 8: ADC2 LSB  
Byte 9: ADC2 MSB

Byte 10: ADC3 LSB  
Byte 11: ADC3 MSB  
Byte 12: ADC4 LSB  
Byte 13: ADC4 MSB  
Byte 14: ADC5 LSB  
Byte 15: ADC5 MSB

Die Bytes 1...3 geben den Zustand der digitalen I/O-Leitungen wieder. Ist eine Leitung HIGH (5V) so ist das zugehörige Bit gesetzt, andernfalls ist es gelöscht. Dieses gilt unabhängig davon, ob die Leitung als Eingang oder als Ausgang arbeitet.

Die Bytes 4...15 liefern paarweise die 12-Bit Datenworte des A/D-Wandlers, der die Spannungen am Port C misst. Der Spannungswert ergibt mit Hilfe der verwendeten Referenzspannung zu:

$$\text{Spannung} = VREF * (\text{MSB} * 256 + \text{LSB}) / 1023$$

Hinweise zur Verwendung des Protokolls

Zum Datenaustausch wird zunächst ein Request-Datenblock (Byte.array) mit einer Länge von genau sieben Bytes definiert, der mit ALLEN Informationen belegt sein muss, die zum Gerät geschickt werden sollen. Dann überträgt man diesen Block über die Schnittstelle. Anschliessend müssen die 15 Antwort-Bytes von der Schnittstelle eingelesen und nach Bedarf ausgewertet werden. Ein neuer Request-Block sollte erst dann ausgegeben werden, wenn alle 15 Bytes als Antwort auf den vorhergehenden Request empfangen wurden.

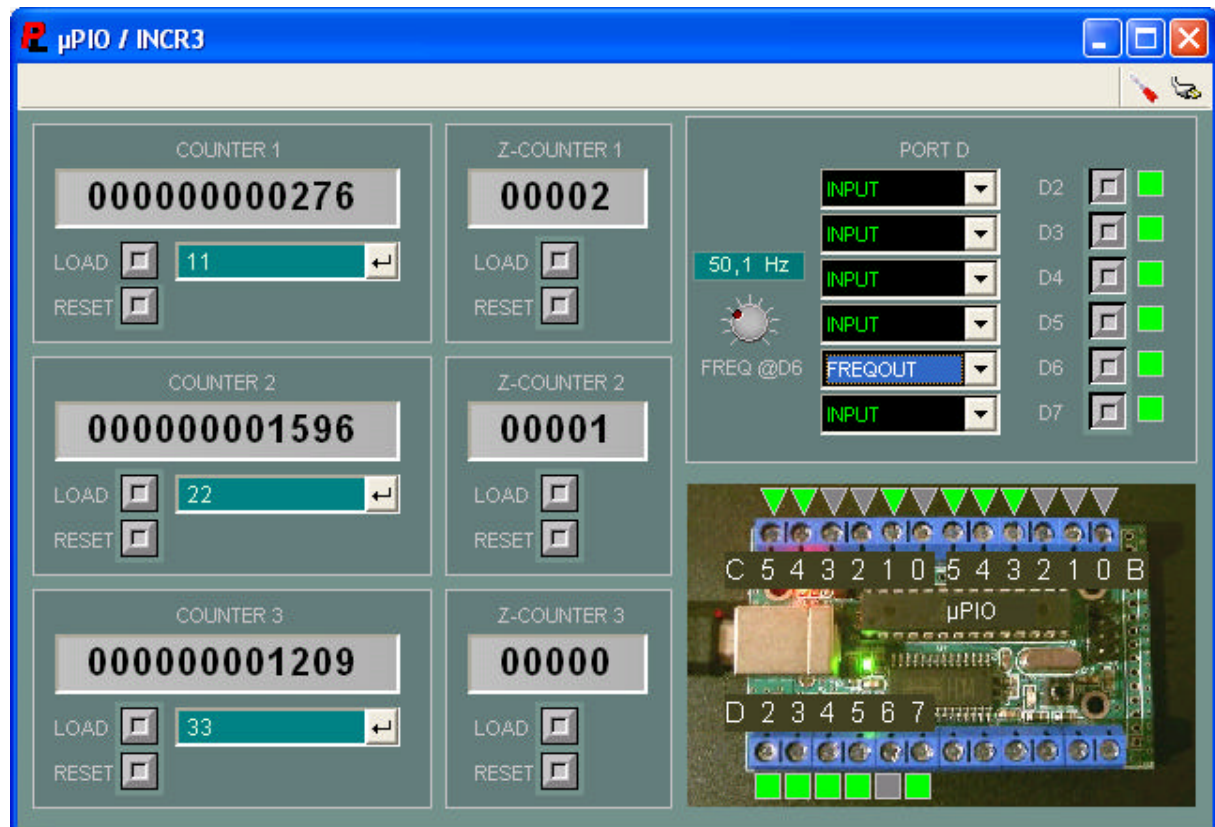
## µPIO / INCR3

Das µPIO / INCR3 erlaubt den Anschluss von drei Inkrementalgebern mit TTL(5V)-Pegeln. Die typischen Signale der Inkrementalgeber (A; B) werden mit 32-Bit (Positions-) Zählern erfasst. Jede fallende Flanke am A-Eingang dient als Taktsignal für den zugehörigen Positionszähler. Die Zählrichtung wird durch das Richtungssignal am B-Eingang bestimmt (high = aufwärts / low = abwärts). Die drei Zählerkanäle verfügen über Freigabe-Eingänge (EN; Enable), die den jeweiligen Zähler freigeben oder sperren.

Die maximale Zählfrequenz liegt bei 20 KHz (Tastverhältnis 1:1).

Sofern vorhanden kann ein Index-Signal (Z) verwendet werden, welches bei jeder vollen Umdrehung des Gebers einen kurzen Puls abgibt. Dabei wird der Positionszähler durch eine Flanke am Z-Eingang auf Null zurückgesetzt, so dass die Positionszählung dann synchron und relativ zur letzten Z-Flanke erfolgt. Ausserdem taktet das Z-Signal einen 16-Bit (Umdrehungs-)Zähler Z , dessen Zählrichtung durch die Zählrichtung des Positionszählers bestimmt wird.

Eine kostenlose PC-Software steht zur Verfügung. Diese ermöglicht neben der Bedienung auch die einfache Datenabfrage per DDE aus anderen Programmen.



Die Zählersignale sind den Anschlüssen von Port B und Port C wie folgt zugeordnet:

#### Zähler 1

Port C0	Eingang	1A	CLK
Port C1	Eingang	1B	DIR
Port B0	Eingang	1Z	/RES
Port B3	Eingang	1EN	EN

#### Zähler 2

Port C2	Eingang	2A	CLK
Port C3	Eingang	2B	DIR
Port B1	Eingang	2Z	/RES
Port B4	Eingang	2EN	EN

#### Zähler 3

Port C4	Eingang	3A	CLK
Port C5	Eingang	3B	DIR
Port B2	Eingang	3Z	/RES
Port B5	Eingang	3EN	EN (Jumper J4 LED entfernen!)

Alle Zähler-Eingänge verwenden TTL 5V-Pegel und verfügen über interne Pullup-Widerstände, die offene Eingänge auf das Potential der positiven Betriebsspannung Vcc ziehen. Dies ermöglicht die z.B. die einfache Beschaltung mit Schaltkontakten gegen Masse.

Port D arbeitet als digitaler I/O-Port. Die Datenrichtung ist für jede Datenleitung einzeln umschaltbar. Port D6 gibt ein Taktsignal aus, sofern dieser auf Ausgang geschaltet ist. Die max. Grundfrequenz von 9 kHz ist durch die Faktoren 1, 2, 3..., 256 teilbar. Für Leitungen die als Eingang arbeiten ist ein Pullup-Widerstand aktivierbar. Die Programmierung erfolgt mit Hilfe von Kommandos, die im Request-Datenblock übertragen werden.

### Datenprotokoll

Die Kommunikation mit dem PC erfolgt per USB. Der USB-Treiber installiert einen virtuellen COM-Port. Der Datenaustausch erfolgt somit wie über eine ‚echte‘ RS232-Schnittstelle.

#### Die Schnittstellenparameter sind: 57600, 8 N, 1

Baud 57600  
8 Datenbits  
Keine Parität  
1 Stopbit

Die Kommunikation erfolgt nach dem Frage-Antwort-Prinzip (Request/Response). Dabei besteht ein Request immer aus genau fünf Datenbytes. Nachdem Empfang dieser fünf Datenbytes sendet die Firmware in jedem Fall 21 Bytes als Antwort an den PC zurück. Die erzielbare Abfragerate (sample rate) liegt bei etwa 150 Samples/Sek.

#### Request: 5 Bytes

Byte 1: Kommando  
Byte 2...5: Parameter

Das Kommando (Byte1) enthält eine Funktionsnummer, mit deren Hilfe verschiedene Geräteaktionen veranlasst werden können. Nicht definierte Funktionsnummern rufen nur den Antwortdatenblock ab, führen aber keine zusätzlichen Aktionen aus. Der Parameter (Byte2...Byte4) muss in jedem Fall mitgesendet werden, auch wenn dieser nur ausgewertet wird, wenn eine Funktion zusätzliche Angaben benötigt. Dieser besteht aus vier Bytes die je nach Bedarf einen 32-Bit-, 16 Bit oder 8 Bit-Wert übertragen. Dabei ist Byte2 das niederwertigste Byte und Byte 5 das höchstwertigste Byte des Parameters.

Folgende Kommandos können im Request übertragen werden:

Dezimal	HEX	CHR	Beschreibung	Parameter
<b>Zähler-Kommandos</b>				
65dez	41h	„A“	Reset Counter 0	beliebig
66dez	42h	„B“	Reset Counter 1	beliebig
67dez	43h	„C“	Reset Counter 2	beliebig
68dez	44h	„D“	Reset Z-Counter 0	beliebig
69dez	45h	„E“	Reset Z-Counter 1	beliebig
70dez	46h	„F“	Reset Z-Counter 2	beliebig
71dez	47h	„G“	Laden Counter 0	INT32 / DWORD
72dez	48h	„H“	Laden Counter 1	INT32 / DWORD
73dez	49h	„I“	Laden Counter 2	INT32 / DWORD
74dez	4Ah	„J“	Laden Z-Counter 0	INT16 / WORD
75dez	4Bh	„K“	Laden Z-Counter 1	INT16 / WORD
76dez	4Ch	„L“	Laden Z-Counter 2	INT16 / WORD
<b>Port D-Kommandos</b>				
88dez	58h	„X“	Frequenzteiler $f=9 \text{ KHz} / (N+1)$	Byte
89dez	59h	„Y“	PORTD = Status / Pullup	Byte
90dez	5Ah	„Z“	DDRD = Datenrichtung	Byte

### Beispiel-Request Byte1...Byte 5: <5A><FC><00><00><00>

<5A> = Kommando „Datenrichtung Port D“

<FC> = %1111 1100 = Bit2...Bit7 gesetzt = D2...D7 Datenrichtung auf Ausgang setzen

Nach dem Empfang von fünf Request-Bytes, antwortet das Gerät stets mit einem Response-Datenblock. Ein neuer Request darf erst nach dem vollständigen Empfang der Antwort gesendet werden.

### Response: 21 Bytes

#### Port-Status:

Byte1: Status Port B Bit 0..5 = B0..B5  
Byte2: Status Port C Bit 0..5 = C0..C5  
Byte3: Status Port D Bit 2..7 = D2..D7

#### Positionszähler:

Byte4: Zähler1 Bit 0...7 (LSB)

Byte5:	Zähler1	Bit 8...15
Byte6:	Zähler1	Bit 16...23
Byte7:	Zähler1	Bit 24...32 (MSB)
Byte8:	Zähler2	Bit 0...7 (LSB)
Byte9:	Zähler2	Bit 8...15
Byte10:	Zähler2	Bit 16...23
Byte11:	Zähler2	Bit 24...32 (MSB)
Byte12:	Zähler3	Bit 0...7 (LSB)
Byte13:	Zähler3	Bit 8...15
Byte14:	Zähler3	Bit 16...23
Byte15:	Zähler3	Bit 24...32 (MSB)

### **Umdrehungszähler (Z):**

Byte16: Zähler1 Bit 8...15 (LSB)  
 Byte17: Zähler1 Bit 0...7 (MSB)

Byte18: Zähler2 Bit 8...15 (LSB)  
 Byte19: Zähler2 Bit 0...7 (MSB)

Byte20: Zähler3 Bit 8...15 (LSB)  
 Byte21: Zähler3 Bit 0...7 (MSB)

Die Bytes 1...3 geben den Zustand der digitalen I/O-Leitungen wieder. Ist eine Leitung HIGH (5V) so ist das zugehörige Bit gesetzt, andernfalls ist es gelöscht. Dieses gilt unabhängig davon, ob die Leitung als Eingang oder als Ausgang arbeitet. Die Bytes 4..15 liefern die 32-Bit Zählerwerte der Positionszähler. Die Bytes 16...21 liefern die 16-Bit Zählerwerte der Umdrehungszähler.

## μPIO / FREQ

Das μPIO / FREQ arbeitet als komfortabler **9-MHz-Frequenzzähler** (Counter) mit zahlreichen Zusatzfunktionen, wie Digital-I/O, PWM-Ausgang, Puls-/Taktgenerator und sechs A/D-Eingängen. Eine kostenlose PC-Software steht zur Verfügung. Diese ermöglicht neben der Bedienung auch die einfache Datenabfrage per DDE aus anderen Programmen.



### Digital-I/O

Für alle Anschlüsse ist eine feste Datenrichtung vorgesehen, die nicht verändert werden kann. Alle Anschlüsse arbeiten mit 5V-TTL-Pegel und sind entsprechend der vorgegebenen Datenrichtung entweder als digital Eingang oder als Digitalausgang nutzbar. Je nach Verwendung und Konfiguration übernehmen die Leitungen aber andere, alternative Funktionen. Die Datenrichtung bleibt aber in jedem Fall erhalten.

### 9-MHz-Frequenzzähler

Die Einspeisung des Zählersignals erfolgt GLEICHZEITIG über die Anschlüsse D2 und D5. Je nach Eingangsfrequenz erfolgt die Frequenzmessung entweder durch eine Impulszählung mit Torzeit (D5;  $F > 250$  Hz) oder durch Flankenmessung (D2;  $F < 250$  Hz).

Als Torzeiten (Gate) für Impulszählung stehen folgende Zeiten zur Verfügung: 2000ms / 1000 ms / 500ms / 100 ms / 50 ms / 20 ms / 10ms. Bei Bedarf kann das Torzeitsignal (Gate) über den Digitalausgang D7 ausgegeben werden.

Bei niedrigen Signalfrequenzen ( $< 250$  Hz) wird zusätzlich die vergangene Zeit ermittelt und ausgegeben, die seit dem letzten Auftreten einer fallenden Flanke am Eingang D2 vergangen ist (elapsed time). Bei niedrigen Frequenzen können somit Frequenzmessung, Zählerfunktion und Zeitmessung parallel verwendet werden. Bei höheren Frequenzen sind nur noch die Frequenzmessung und die Zählerfunktion alternativ nutzbar.

### 32-Bit-Counter

Ohne Torzeitgenerator arbeitet der Frequenzzähler als Counter, der fallende Flanken an D5 aufwärts zählt. Digitaleingang B0 kann per Software als externer RESET-Eingang für den

Zähler konfiguriert werden. Digitaleingang B4 kann per Software als externer ENABLE-Eingang (Freigabe) für den Zähler konfiguriert werden.

### **Generator 1**

Dieser Generator erzeugt ein periodisches Ausgangstaktsignal am Ausgang D6. Dieses Ausgangssignal wird aus einem Eingangstaktsignal abgeleitet.

Das Eingangstaktsignal wird dazu einem internen 8-Bit-Zähler als Taktsignal zugeführt. Durch einen internen Vergleich des 8-Bit-Zählerstandes mit einem Vergleichswert, wird schliesslich das Ausgangssignal D6 gebildet. Je nach Konfiguration kann dieses ein Ausgangstaktsignal mit einstellbarer Frequenz oder mit einstellbarer Pulsbreite (PWM) sein.

Das Eingangstaktsignal kann entweder intern oder extern eingespeist werden.:

Bei interner Taktquelle wird der Systemtakt (18.432000 MHz) mit einem einstellbaren Vorteiler durch die Faktoren 1, 8, 64, 256 oder 1024 heruntergeteilt. Eingang D4 kann als per Software als externer Freigabeeingang für den internen Takt konfiguriert werden.

Bei externer Taktquelle dient Digitaleingang D4 als externer Takteingang, der per Software freigegeben oder gesperrt werden kann.

### **Generator 2**

Dieser Generator ist ein Pulsgenerator, der sein Ausgangssignal an D3 ausgibt. Die Zeitdauer für die High- und Low-Phase des Ausgangssignals ist einstellbar. Die Zeiten stellen ganzzahlige Vielfache von vier möglichen Grundzeiteinheiten (Timebase) dar, die um den Faktor 1...65535 verlängert werden können. Die Grundzeiteinheiten sind 125µs, 500µs, 5ms und 500 ms. Während die Grundzeit für High- und Low-Phase dieselbe ist, können die Faktoren für die High-/Low-Dauer getrennt eingestellt werden.

Der Ausgangspuls von Generator 2 kann fortlaufend (continuous) oder einmalig (single-shot) ausgegeben werden. Ein Einzelimpuls kann per Software ausgelöst werden. Alternativ kann Digitaleingang B1 per Software als Triggereingang konfiguriert werden. Bei fortlaufender Pulsausgabe wird das Ausgangssignal mit dem Triggersignal (B1 oder Software) synchronisiert.

### **Generator 3**

Dieser Generator erzeugt am Digitalausgang D3 eine feste Ausgangsfrequenz von 72.000 KHz und einstellbarer Pulsbreite (8-Bit PWM).

Die Erzeugung der PWM erfolgt wie bei Generator 1 durch den Vergleich eines internen Zählerstandes mit einem Vergleichswert.

## **Anschlussbelegung**

Die Anschlüsse sind wie folgt belegt:

B0	Digital Input (Counter RESET)
B1	Digital Input (Sync/Trigger Generator 2)
B2	Digital Output
B3	Digital Output (Generator 3 PWM)
B4	Digital Input (Counter enable)
B5	Digital Output (LED)
D2	Digital Input (Frequenz-Eingang 0-250Hz)

D3 Digital Output (Generator 2)  
 D4 Digital Input (EXT CLK ENABLE)  
 D5 Digital Input (Frequenz-Eingang 250 Hz - 8MHz)  
 D6 Digital Output (Generator 1)  
 D7 Digital Output (Gate Clock Output)

C0...C5 Digital Inputs / ADC 0...5

Alle Eingänge von Port B und Port D sind mit Pullup-Widerständen auf Vcc-Potenzial gezogen. Port C kann als Digitaleingang (ohne Pullups) verwendet werden.

Die sechs Spannungen am Port C können alternativ mit dem internen 10-Bit-A/D-Wandler (ADC) eingelesen werden. Die Referenzspannung für den ADC kann dabei aus drei umschaltbaren Quellen bezogen werden:

Externe Referenzspannung:

Die Referenzspannung wird ggf. über den Anschluss VREF eingespeist.

$0 < V_{ref} < V_{cc}$  ist dabei einzuhalten.

Interne Referenz VCC

Als Referenzspannung dient die positive Versorgungsspannung (also ca. 5V).

Interne Referenz 1.1 V

Eine Referenzspannung von 1.1V wird intern erzeugt und dient als Referenz.

## Datenprotokoll

**Die Schnittstellenparameter sind: 57600, 8 N, 1**

Baud 57600

8 Datenbits

Keine Parität

1 Stopbit

Die Kommunikation erfolgt nach dem Frage-Antwort-Prinzip (Request/Response). Dabei besteht ein Request immer aus genau 10 Datenbytes. Nachdem Empfang dieser 10 Datenbytes sendet die Firmware in jedem Fall 33 Bytes als Antwort an den PC zurück. Ein neuer Request darf erst gesendet werden, wenn die 33 Antwortbytes empfangen wurden. Die erzielbare Abfragerate (sample rate) liegt bei etwa 50 Samples/Sek.

### Request: 10 Bytes

Der Request-Datenblock hat eine Länge von 10 Bytes und überträgt sämtliche Daten zum Gerät, die für die Steuerung relevant sind:

#### Byte 1 Torzeit und Digital Out

Die Bit 0..2 selektieren die gewünschte Torzeit für die Frequenzmessung:

Bit	Hex	Dez	Torzeit
210			
000	00h	0	2 s
001	01h	1	1s

010	02h	2	500 ms
011	03h	3	100 ms
100	04h	4	50 ms
101	05h	5	20 ms
110	06h	6	10 ms
111	07h	7	AUS / COUNTER

Die Bits 3...7 sind der Reihe nach den digitalen Ausgängen B2, B3, B5, D3 und D6 zugeordnet und setzen deren Ausgangsstatus bzw. aktivieren die alternative Funktion des jeweiligen Anschluss, wie z.B. Generator 2 am Ausgang D3.

#### **Byte 2 Vergleichswert für Generator 1**

Dieses Byte stellt den 8-Bit-Vergleichswert für Generator 1 ein und variiert je nach Modus die Ausgangsfrequenz bzw. die Pulsbreite des Ausgangssignal (D6).

#### **Byte 3 Vergleichswert für Generator 3**

Dieses Byte stellt den Vergleichswert für Generator 3 ein und variiert die Pulsbreite des Ausgangssignal (B3).

#### **Byte 4 Modus**

In Byte 4 sind folgende Funktionen bitweise kodiert:

- Modus und Taktquelle für Generator 1
- Zeitbasis für Generator 2
- Counter Reset per Software

Der Wert für Byte 4 ergibt sich aus der Addition der nachstehenden HEX oder Dezimalwerte, je nach gewünschter Funktion.

Die **Bits 0...2** selektieren die Taktquelle **für Generator 1**.

Bit [210]	Hex	Dez	Taktquelle
000	00h	0	KEINE
001	01h	1	CLK 18.432000 MHz
010	02h	2	CLK / 8 2.304000 MHz
011	03h	3	CLK / 64 288.000 KHz
100	04h	4	CLK / 256 72.000 KHz
101	05h	5	CLK / 1024 18.000 KHz
110	06h	6	EXTERNER CLOCK (D4), fallende Flanke
111	07h	7	EXTERNER CLOCK (D4), steigende Flanke

**Bit 3** setzt den Zählerstand des (Frequenz-)zählers auf Null zurück (**Counter Software Reset**), wenn das Bit gesetzt ist.

Bit [3]	Hex	Dez	
	08h	8	Counter Reset

**Bit 4 und Bit 5** wählen den **Modus von Generator 1**

Bit [54]	Hex	Dez	Modus Generator 1
00	00h	0	Normal
01	10h	16	PWM
10	20h	32	CTC

**Modus: Normal**

Der Generator arbeitet als Frequenzteiler, der die Frequenz des Eingangstaktsignals zunächst durch 2 und anschliessend durch 256 teilt.

$$F_{out} = (F_{in} / 2) / 256$$

Feste Ausgangsfrequenzen bei internem Takt:

36.000 kHz  
4.500 kHz  
562,5 Hz  
140,625 Hz  
35,15625 Hz

**Modus: PWM**

Der Generator arbeitet als Frequenzteiler, der die Frequenz des Eingangstaktsignals zunächst durch 2 und anschliessend durch 255 teilt. Die Pulsbreite wird durch den Vergleichswert bestimmt.

$$F_{out} = (F_{in} / 2) / 255$$

PWM-Frequenzen bei internem Takt:

36.141 kHz  
4.518 kHz  
564,7 Hz  
141,17 Hz  
35,294 Hz

**Modus: CTC**

Der Generator arbeitet als einstellbarer Frequenzteiler, der die Frequenz des Eingangstaktsignals zunächst durch 2 und anschliessend durch den Vergleichswert+1 teilt.

$$F_{out} = (F_{in} / 2) / (CMP1 + 1)$$

Frequenzbereiche bei internem Takt:

36.141 kHz – 9.216 MHz  
4.518 kHz – 1.152 MHz  
564,7 Hz - 144.0 kHz  
141,17 Hz - 36.0 kHz-  
35,294 Hz – 9.0 kHz

**Modus: Fast PWM**

Der Generator arbeitet als Frequenzteiler, der das Eingangstaktsignal durch 256 teilt. Die Pulsbreite wird durch den Vergleichswert bestimmt

$$F_{out} = F_{in} / 256$$

PWM-Frequenzen bei internem Takt:

72.000 kHz  
9.0 kHz  
1.125 kHz  
281,25 Hz  
70,3125 Hz

**Bit 6 und Bit 7** stellen die **Zeitbasis von Generator 2** ein.

Bit	Hex	Dez	Zeitbasis Generator 2
-----	-----	-----	-----------------------

[76]

00	00h	0	125 µs
01	40h	64	500 µs
10	80h	128	5 ms
11	C0h	192	500 ms

**Byte 5...6 TH; Generator 2 ; Pulsdauer Dauer HIGH ( INT16 / WORD )**

**Byte 7...8 TL; Generator 2; Pulsdauer Dauer LOW ( INT16 / WORD )**

Vier Bytes bilden jeweils einen ganzzahligen 16-Bit Wert, welche die High / Low-Dauer und damit das Tastverhältnis und die Frequenz des Ausgangssignals von Generator 2 (D3) bestimmen. Der Zeitwert ergibt sich aus der Multiplikation mit der gewählten Zeitbasis von Generator 2.

$T_{high} [s] = TH * \text{Zeitbasis}$

$T_{low} [s] = TL * \text{Zeitbasis}$

Mit  $TH = TL = 1$  und einer Zeitbasis von  $125\mu s$  ergibt sich die maximale Ausgangsfrequenz von 4 kHz.  $f = 1 / ((TL + TH) * \text{Zeitbasis})$

Mit  $TH = TL = FFFFh$  und einer Zeitbasis von  $500ms$  ergibt sich Ausgangssignal mit einer maximalen Periodendauer von  $65535$  Sekunden, also mehr als 18 Stunden.  
 $T = (TL + TH) * \text{Zeitbasis}$

### **Byte 9 Konfiguration und Digital Out (Flags)**

Die Bits in diesem Byte dienen als Flags für verschiedene Konfigurationen und Funktionen.

- Bit 0 Ist Bit 0 gesetzt, arbeitet B0 als RESET-Eingang für den (Frequenz-)Zähler (Counter)
- Bit 1 Ist Bit 1 gesetzt, arbeitet B1 SYNC/TRIGGER-Eingang für Generator 2
- Bit 2 Ist Bit 2 gesetzt, arbeitet B4 als ENABLE-Eingang für den (Frequenz-)Zähler (Counter)
- Bit 3 Bit 3 bewirkt die Umschaltung des Modus von Generator 2: Continuous / Single-shot
- Bit 4 Bit 4 dient als SYNC/TRIGGER für Generator 2 (Software)
- Bit 5 Bei **externer** Taktquelle für Generator 2:
  - Bit 5 gesetzt: Taktsignal von Eingang D4 freigegeben.
  - Bit 5 gelöscht Taktsignal von Eingang D4 gesperrt.Bei **interner** Taktquelle für Generator 2:
  - Bit 5 gesetzt: Eingang D4 dient als Freigabesignal für den internen Takt.
  - Bit 5 gelöscht Interner Takt immer freigegeben (unabhängig von Eingang D4)
- Bit 6 Bit 6 aktiviert die Ausgabe des Torzeit-Taktsignals am Ausgang D7.
- Bit 7 Bit 7 schaltet den digitalen Ausgang D7 ein/aus.

### **Byte 10 ADC Referenzumschaltung (VREF SELECT)**

**Bit 0..1** wählen die **Referenzspannung** für die ADC-Spannungsmessung (C0...C5)

Bit	Hex	Dez	Modus Generator 1
10			
00	00h	0	Externe Referenzspannung an VREF
01	01h	1	Interne Referenzspannung (Vcc)
10	02h	2	Nicht definiert
11	03h	3	Interne Referenzspannung 1.1 Volt
Bit 2..7	Reserviert		Auf Null setzen



## Response: 33 Bytes

Nach dem Empfang von 10 Request-Bytes antwortet das Gerät mit 33 Response-Bytes:

**Byte 1...4**      **Flankenmessung T1 ( INT32 / DWORD )**  
**Byte 5...8**      **Flankenmessung T2 ( INT32 / DWORD )**  
**Byte 9...12**     **Flankenmessung T3 ( INT32 / DWORD )**

Vier Bytes bilden jeweils einen ganzzahligen 32-Bit Wert (LSB first). Die drei Datenworte T1, T2 und T3 liefern das Ergebnis der Flankenmessung. Bei einer fallenden Flanke am Eingang D2 wird der Zählerstand eines freilaufenden 32-Zählers abgelesen, der intern mit 72 kHz ( T = 13,8µs ) getaktet ist.

Das Ergebnis dieser Ablesung befindet sich im Datenwort T2. Der Zählerstand der vorhergehenden Flanke wird zuvor in das Datenwort T1 verschoben. **T1 und T2 enthalten somit die Zählerstände von zwei aufeinander folgenden, fallenden Flanken.**

Per Software lässt sich daraus leicht die Periodendauer T und Frequenz F wie folgt ermitteln:

$$T = (T2 - T1) * 13,8 \mu s$$

$$F = 1 / T$$

Da der interne Zähler bei einem Wert von \$FFFFFFFF überläuft und seine Zählung dann bei Null beginnt, sollte vor der Berechnung die Bedingung  $T2 \geq T1$  geprüft werden.

Das Datenwort T3 enthält zusätzlich den aktuellen Zählerstand und ist unabhängig von der Flankenerkennung.

Die Differenz (T3 – T2) liefert somit ein Mass für die verstrichene Zeit, seit dem letzten Auftreten einer fallenden Flanke an D2. (Elapsed Time).

$$T_{\text{elapsed}} = (T3 - T2) * 13,8 \mu s$$

Auch hier sollte  $T3 \geq T2$  geprüft werden.

## **Byte 13...16**      **Frequenzzähler / Counter CNT ( INT32 / DWORD )**

Vier Bytes bilden einen ganzzahligen 32-Bit Wert (LSB first). Das Datenwort enthält den Zählerstand eines 32-Bit-Zählers, der durch den Frequenzeingang D5 getaktet aufwärts zählt. Sofern eine Torzeit aktiv ist, enthält das Datenwort den Zählerstand bei Ablauf der Torzeit. Mit Ablauf der Torzeit wird der Zähler auf Null zurückgesetzt. (Berechnung der Frequenz: siehe Byte 20)

Bei deaktivierter Torzeit enthält das Datenwort den aktuellen Zählerstand (Counter).

## **Byte 17**                      **Digitalstatus Port B (Byte)**

Die Bits 0..5 zeigen den Digitalstatus der Anschlüsse B0...B5 an.

## **Byte 18**                      **Digitalstatus Port C (Byte)**

Die Bits 0..5 zeigen den Digitalstatus der Anschlüsse C0...C5 an.

## **Byte 19**                      **Digitalstatus Port D (Byte)**

Die Bits 2..7 zeigen den Digitalstatus der Anschlüsse D2...D7 an.

Die Erfassung des Digitalstatus ist unabhängig von der Datenrichtung.

## **Byte 20**                      **Torzeit TZ [1/100 s]**

Das Byte liefert die Torzeit (in 1/100 s), die dem Zählerstand CNT zu Grunde liegt. Per Software lässt sich aus der Torzeit TZ und dem Zählerstand CNT leicht die Frequenz am Eingang D5 ermitteln:

$$F [\text{Hz}] = \text{CNT} / ( \text{TZ} / 100 )$$

Ist keine Torzeit aktiv (Counter) ist TZ=0 und somit keine Frequenz zu ermitteln.

**Byte 21..22     ADC0 (Word)**  
**Byte 23..24     ADC1 (Word)**  
**Byte 25..26     ADC2 (Word)**  
**Byte 27..28     ADC3 (Word)**  
**Byte 29..30     ADC4 (Word)**  
**Byte 31..32     ADC5 (Word)**

Jeweils zwei Byte liefern ein 16 Bit Datenwort (LSB first). Das Datenwort enthält das Ergebnis der A/D-Konvertierung (10 Bit ADC; 000h....3FFh). Ein Spannungswert kann mit Hilfe der Referenzspannung Vref leicht errechnet werden:

$$U_0 [\text{Volt}] = \text{ADC0} / 3\text{FFh} * V_{\text{ref}}$$

**Byte 33                     Reserviert**

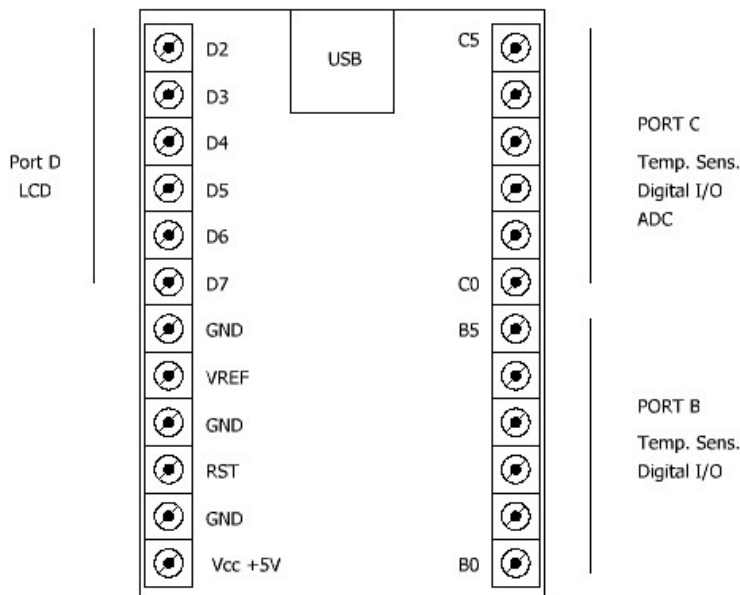
Das Byte 33 ist reserviert und liefert derzeit den Wert 13.

## µPIO / TEMP12

Das µPIO / TEMP12 ist für den Betrieb von bis zu zwölf digitalen Temperatursensoren vom Typ Dallas DS18B20 ausgelegt. Die Werte der zwölf Kanäle werden per USB zum PC übertragen. Der Anschluss der Sensoren erfolgt über die Anschlussklemmen B0...B5 und C0...C5. An jeden dieser Kanäle kann EIN Temperatursensor angeschlossen werden.

Freie Kanäle, die nicht für Temperatursensoren verwendet werden, können für alternative Aufgaben konfiguriert werden:

- DS18B20 Temperatursensor
- Digitaleingang
- Digitalausgang
- Alarmausgang
- Sensoreingang (ADC) (nur C0...C5)



Die Konfiguration des Gerätes erfolgt mit Hilfe der mitgelieferten Software. Diese ermöglicht die Einstellung der Kanäle. Die Kanaleinstellungen werden mit Hilfe der Konfigurationssoftware zum Gerät übertragen und dort temporär oder dauerhaft abgelegt. Somit ist das µPIO / Temp12 als PC-Interface geeignet, kann aber auch für den Betrieb ohne PC (Stand-Alone) verwendet werden.

Für jeden analogen Kanal (Temperatur- und Sensoreingänge) können zwei Grenzwerte definiert werden, die den Eingangswert als „HOCH“ und „TIEF“ einstufen. Diese Einstufung dient zur Ansteuerung von Kanälen, die als Alarmausgang konfiguriert sind. So kann ein Alarmausgang z.B. ein Steuersignal ausgeben, welches etwa einen Lüfter einschaltet, sobald eine Temperatur den Grenzwert „HOCH“ überschreitet. Ausserdem erscheint die Anzeige „HOCH“ bzw. „TIEF“ ggf. auch auf einem optionalen LCD-Display (siehe LCD-Option).

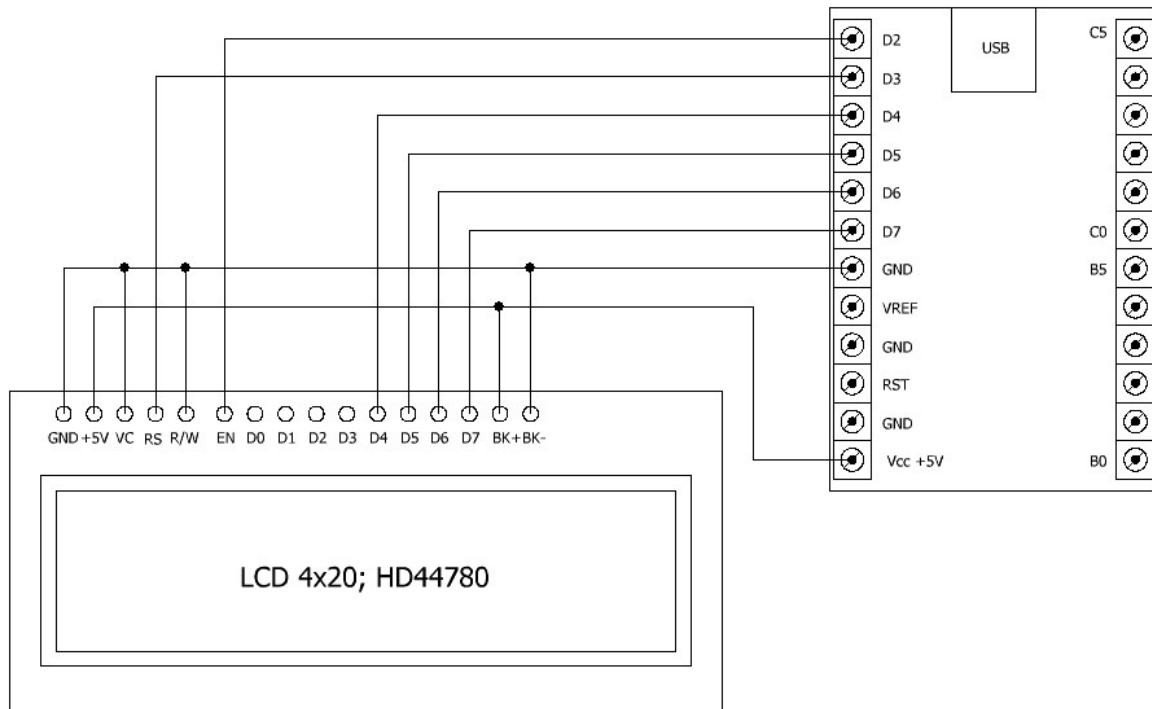
## LCD-Option

Die Anschlüsse D2...D7 des  $\mu$ PIO /TEMP12 erlauben den Anschluss eines Standard-LCD-Textdisplays (HD44780) mit 4x20 Zeichen. Andere Displayformate werden NICHT unterstützt. Auf dem Display werden die Werte und Zustände der Datenkanäle automatisch angezeigt.

Der Anschluss des Displays erfolgt am Port D mit folgender Signalbelegung:

$\mu$ PIO / TEMP12 PORT D	LCD 4x20 Zeichen HD44780
D2	EN
D3	RS
D4	D4
D5	D5
D6	D6
D7	D7

Die übrigen Datenleitungen (D0...D3) **des LCD** bleiben unbeschaltet. Die Leitung R/W muss auf Masse-Potential (GND) geschaltet werden.



Anschlussbild LCD-Textdisplay 4x20 (HD44780)

## Kanalkonfiguration - DS18B20 Temperatursensor

Der Anschluss von bis zu 12 Temperatursensoren ist an den Klemmen B0...B5 und C0...C5 möglich. Mit der Konfigurationssoftware muss der Kanal entsprechend konfiguriert werden. Die Temperatursensoren sind für Temperaturen von -55 °C und 125° C geeignet. Sofern eine lineare Umrechnung in andere Temperatureinheiten gewünscht ist, (z.B. °F statt °C) so gibt man den Umrechnungsfaktor und Offset in die Felder für die Sensoranpassung ein.

Beispiel – Umrechnung °C in °F:

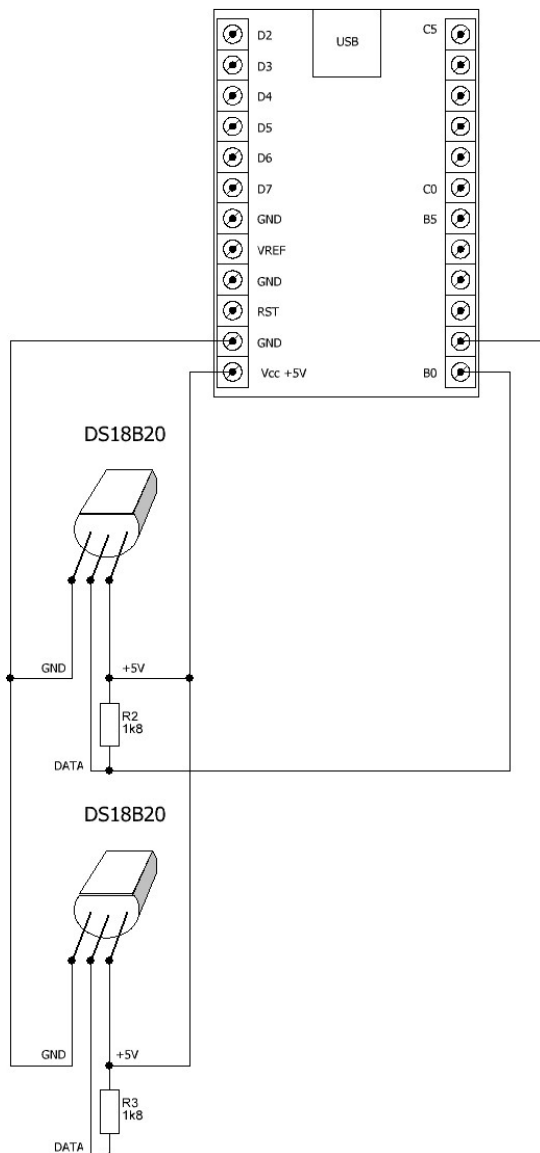
Einheit = „°F“

Faktor = 1,8

Offset = 32

Eine solche Skalierung des Messwerts verändert lediglich die Darstellung des Temperaturwertes. Dieses verändert jedoch NICHT den physikalischen Messbereich (-55 °C bis 125° C) des Sensors.

Die Eingabe der Grenzwerte für „HOCH“ und „TIEF“ erfolgt in der Einheit, in der Sie den Sensor skaliert haben.



Anschlussbild DS18B20 an Kanal B0 und B1

### **Kanalkonfiguration – ADC Sensoreingang**

Diese Konfiguration ist nur für die Kanäle C0...C5 möglich. Ein Analog-Digital-Wandler (ADC) erfasst eine Eingangsspannung, die z.B. ein Sensor liefert. Dazu vergleicht er intern die Eingangsspannung mit einer Referenzspannung VREF. Als Referenzspannungsquelle dient dabei entweder die (USB-) Betriebsspannung (Vcc; 5V), eine interne Referenz (1,1V) oder eine externe Referenzspannung, die über den Anschluss REF eingespeist wird. Die Referenzspannungsquelle ist für jeden Sensoreingang einzeln konfigurierbar, d.h. umschaltbar.

Die maximal zulässige Referenzspannung ist die Betriebsspannung Vcc! Die Eingangsspannung muss zwischen 0V und VREF liegen, damit sie vom AD-Wandler erfasst werden kann. Spannungen < 0 Volt und Spannungen > Vcc sind an allen Eingängen unzulässig!

Bei korrekter Konfiguration der Referenzspannung liefert der ADC Messwerte in der Einheit Volt. Die Sensoranpassung erlaubt es, wie schon bei den Temperaturkanälen, die gemessene Eingangsspannung in eine andere Einheit linear umzurechnen.

Beispiel – Umrechnung Spannung (mit VREF = 5V) in Prozent

Einheit = „%“

Faktor = 20

Offset = 0

So ergibt sich bei einer Eingangsspannung von 5V eine interne Darstellung von  $5(V) * 20 = 100\%$ . Die Umskalierung verändert NICHT den physikalischen Bereich der Eingangsspannung von 0 V bis VREF, sondern nur die interne Darstellung. Die Anpassung des Messbereichs an die Ausgangsspannung des Sensors muss durch eine geeignete Beschaltung des Eingangs erfolgen (Spannungsteiler; Verstärker; o.ä.)

### **Kanalkonfiguration – Alarmausgang**

In dieser Konfiguration arbeitet der Kanal als digitaler (Schalt-) Ausgang. Im eingeschalteten Zustand „EIN“ liefert der Ausgang einen Highpegel (5V), im ausgeschalteten Zustand „AUS“ liefert der Ausgang einen Lowpegel (0V). Der Ausgang ist mit max. 20mA belastbar.

Ein Kanal der als Alarmausgang konfiguriert ist, wechselt seinen Schaltzustand EIN / AUS selbsttätig. Ist eine bestimmte BEDINGUNG erfüllt so schaltet der Ausgang EIN, andernfalls schaltet er AUS. Als Bedingungen dienen die Zustände „HOCH“ und „TIEF“ anderer Messkanäle (Temperatur oder ADC). Um ein „Flackern“ der Alarmausgänge zu vermeiden, wenn der Messwert leicht um den Grenzwert schwankt, ist eine Hysterese von ca. +/- 2 LSB fest vorgesehen.

### **Kanalkonfiguration – Digitalausgang**

In dieser Konfiguration arbeitet der Kanal als digitaler (Schalt-) Ausgang. Im eingeschalteten Zustand „EIN“ liefert der Ausgang einen Highpegel (5V), im ausgeschalteten Zustand „AUS“ liefert der Ausgang einen Lowpegel (0V). Der Ausgang ist mit max. 20mA belastbar. Das Umschalten erfolgt PC gesteuert.

### **Kanalkonfiguration – Digitaleingang**

In dieser Konfiguration arbeitet der Kanal als digitaler Eingang. Der eingeschaltete Zustand „EIN“ wird durch einen Highpegel (5V) signalisiert, der ausgeschaltete Zustand „AUS“ durch einen Lowpegel (0V). Offene (unbeschaltete) Eingänge sind mit einem internen Pullup-Widerstand auf Vcc Potential gelegt. So lässt sich z.B. leicht der Zustand eines Schalters erfassen, der die Eingangsklemme im geschlossenen Zustand mit Masse (GND) verbindet.

Bei Verwendung eines optionalen LCD-Displays werden die Zustände der digitalen Kanäle (Digitaleingang, Digitalausgang, Alarmausgang) als „EIN“ bzw. „AUS“ angezeigt.

## Messen von Differenzen zwischen Sensoren

Da in der Praxis häufig Differenzen zwischen Sensoren ausgewertet werden müssen (z.B. Aussentemperatur – Innentemperatur) haben wir das  $\mu$ PIO / TEMP12 mit vier zusätzlichen (virtuellen) Messkanälen ausgestattet, die per Berechnung die Differenz aus zwei echten Messkanälen bilden. Folgende Differenzen können gebildet werden:

- DIFF1 = Messwertkanal C0 – Messwertkanal C1
- DIFF2 = Messwertkanal C0 – Messwertkanal C2
- DIFF3 = Messwertkanal C0 – Messwertkanal C3
- DIFF4 = Messwertkanal C4 – Messwertkanal C5

Damit die Differenzbildung sinnvoll ist, müssen die beiden Messkanäle, die zur Differenzbildung dienen sollen identisch konfiguriert werden. Ist dieses der Fall, so wird die zugehörige Differenz automatisch gebildet und die dabei Konfiguration der echten Messkanäle verwendet. Lediglich die Grenzwerte HOCH und TIEF sind für Differenzkanäle konfigurierbar und können ebenso wie die Grenzwerte echter Kanäle als Bedingung für Alarmausgänge verwendet werden.

## Übertragen der Konfigurationseinstellungen zum Gerät:

Vor dem eigentlichen Betrieb des Gerätes muss die Konfigurationseinstellungen zum Gerät übertragen werden. Dies geschieht mit Hilfe der Konfigurationssoftware. Die Konfiguration kann wahlweise in den flüchtigen RAM-Speicher oder in den nicht-flüchtigen EEPROM-Speicher des Gerätes übertragen werden. Bei einem Reset des Gerätes liest das Gerät die Konfiguration aus dem nicht-flüchtigen EEPROM-Speicher. Eine Konfiguration, die in den flüchtigen RAM-Speicher übertragen wurde geht dabei verloren. Ein Reset erfolgt beim Einschalten der Versorgungsspannung oder durch einen LOW-Impuls am Reset Anschluss (/RST). Ein Reset erfolgt auch, wenn der PC eine Datenverbindung mit dem Gerät aufbaut. Dieses kann durch das Entfernen des Jumpers J1 SOFT-RESET verhindert werden. Beim Übertragen einer Konfiguration wird eine geräteeigene Uhr mit der Systemzeit des PC synchronisiert. Diese dient der Anzeige von Datum und Uhrzeit auf dem optionalen LCD-Display. Da die Uhrzeit technisch bedingt bei einem Reset verloren geht, erfolgt nach einem Reset keine Uhrzeitanzeige mehr. Für die Uhranzeige ist eine ununterbrochene Spannungsversorgung erforderlich.

## Datenprotokoll

### Die Schnittstellenparameter sind: 57600, 8 N, 1

Baud 57600  
8 Datenbits  
Keine Parität  
1 Stopbit

Die Kommunikation erfolgt nach dem Frage-Antwort-Prinzip (Request/Response). Dabei besteht ein Request immer aus genau 21 Datenbytes. Nachdem Empfang dieser 21 Datenbytes sendet die Firmware in jedem Fall eine Antwort an den PC zurück. Ein neuer Request darf erst gesendet werden, wenn die Antwort vollständig empfangen wurden.

### Request: 21 Bytes

Der Request-Datenblock hat eine Länge von 21 Bytes

Byte 1: Immer 66 dez = chr(„B“)  
Byte 2...Byte 13: Digitaler Ausgangsstatus B0...B5, C0...C5





## µPIO / ABADUINO

Das Board ähnelt in Teilen den Arduino USB Schaltplänen (<http://www.arduino.cc>). Der Bootloader wurde für die geänderte Taktfrequenz von 18.432 MHz modifiziert. Die Bootloader-Source ist im Software-Download enthalten und öffentlich gemacht. Sie finden diese Zusammen mit der Installationsanleitung im Ordner SOFTWAREABADUINO.

```
=====  
Installation µPIO / ABADUINO in Arduino-Software:  
=====
```

Schliessen Sie ggf. zunächst alle Arduino-Fenster.

- 1.) Finden Sie die Datei BOARDS.TXT in Ihrer Arduino-Installation. (z.B. C:\arduino-0021\hardware\arduino\boards.txt)
- 2.) Öffnen Sie die Datei mit einem Texteditor (WordPad).
- 3.) Kopieren Sie den nachstehenden Abschnitt und fügen Sie diesen GANZ UNTEN in die Datei BOARDS.TXT ein. Speichern und schliessen Sie die Datei.

```
#####
```

```
diecimila.name=uPIO /ABADUINO  
  
diecimila.upload.protocol=stk500  
diecimila.upload.maximum_size=14336  
diecimila.upload.speed=19200  
  
diecimila.bootloader.low_fuses=0xff  
diecimila.bootloader.high_fuses=0xdd  
diecimila.bootloader.extended_fuses=0x00  
diecimila.bootloader.path=ABADUINO  
diecimila.bootloader.file=ATmegaBOOT_168_pro_18M432.hex  
diecimila.bootloader.unlock_bits=0x3F  
diecimila.bootloader.lock_bits=0x0F  
  
diecimila.build.mcu=atmega168  
diecimila.build.f_cpu=1843200L  
diecimila.build.core=arduino
```

```
#####
```

- 4.) Finden und öffnen Sie den Ordner BOOTLOADERS in Ihrer Arduino-Installation. (z.B. c:\arduino-0021\hardware\arduino\bootloaders)
- 5.) Kopieren Sie den ORDNER ABADUINO (der auch diese Anleitung enthält) und fügen Sie diesen ORDNER in den Ordner BOOTLOADERS Ihrer Arduino-Installation ein.

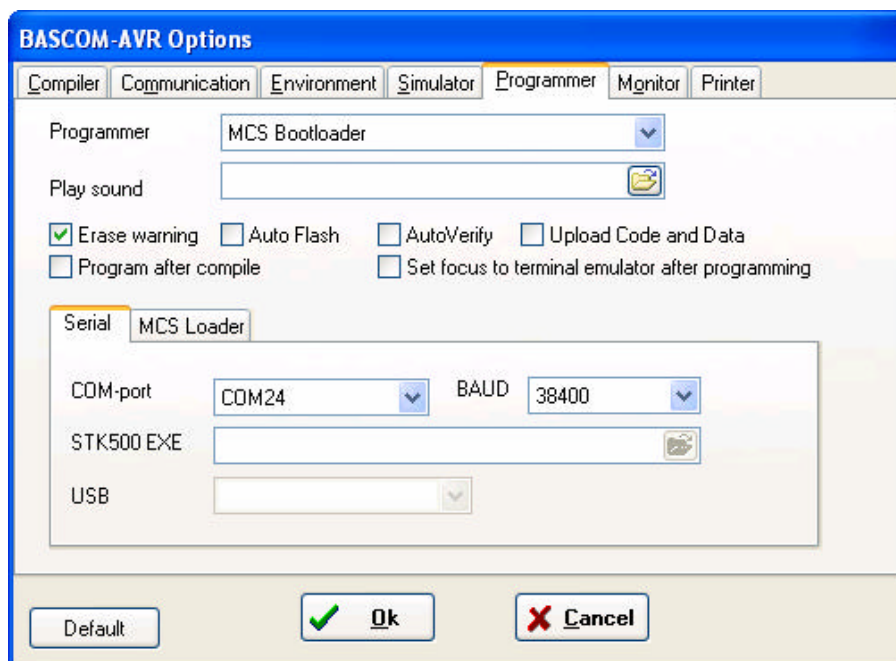
Sie können nun Ihre Arduino-Software starten, das Board auswählen und in gewohnter Weise mit Arduino einsetzen.

## µPIO / MCS

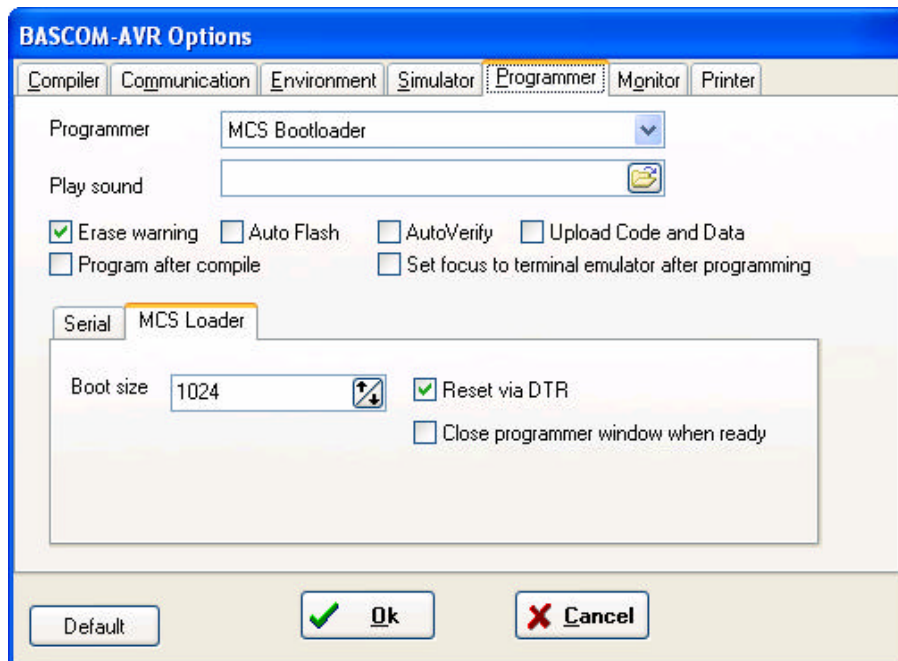
Mit der Firmwareoption µPIO / MCS ist das Board in Verbindung mit BASCOM AVR (MCS Electronics) sofort einsatzbereit. Programme die Sie selbst mit BASCOM AVR entwickelt haben, können bequem per USB auf das Board übertragen werden. Es ist also kein Hardware-Programmiergerät mehr erforderlich. Den MCS-Bootloader haben wir bereits für Sie installiert. Dieser benötigt etwa 2KByte des 16KByte grossen Flash-Speichers.

Bitte arbeiten Sie mit der neuesten Version von BASCOM AVR. Ein kostenloses Update und eine Demoversion gibt auf der Internetseite von MCS.

Zum Übertragen Ihrer Programme stellen Sie in BASCOM AVR unter OPTIONS->PROGRAMMER den MCS-Bootlader ein.



Der COM-Port wird Ihnen vom USB-Gerätetreiber bereit gestellt. (Virtueller COM-Port; VCP). Schauen Sie im Gerätemanager nach welche COM-Port-Nummer vergeben wird. Die Baudrate des Bootloaders ist 38400 Baud. Alle Software-Optionen, wie z.B. ‚Reset via DTR‘ sind damit nutzbar.



Weitere Informationen:

<http://www.mcselec.com>

[AN #143 - MCS Bootloader](#)

## μPIO / Clear

Sie erhalten das μPIO-Board bestückt mit einem unprogrammierten ATmega168-20.  
Die Programmierung muss mit einem handelsüblichen ISP-Programmer erfolgen. Dieser gehört NICHT zu Lieferumfang. Das Board besitzt einen 6-poligen ISP-Anschluss.

Für die Fusebits empfehlen wir folgende Grundeinstellung:

Low Fuse:0xFF

High Fuse:0xDF

Extended Fuse: Fuse:0xF9

### **Programmer-Empfehlung:**

Den „mySmartUSB light“ von myAVR haben wir erfolgreich mit dem μPIO-Board getestet und ist als Zubehör erhältlich. Ein passendes 6-poliges ISP-Kabel liegt dem Programmer bei.

