



# Bedienungsanleitung

# PWM Generator Version 2

© 2016 measX GmbH & Co. KG  
Trompeterallee 110,  
41189 Mönchengladbach, Deutschland  
Telefon +49 (0) 21 66 / 95 20 – 0

Email: [info@measx.de](mailto:info@measx.de)

Web: <http://www.measx.com>

*Autor: Heinz Rottmann  
Grafiken: Robert Puttler*

*Version 1.98*

## 1 Inhalt

2	Übersicht .....	4
2.1	Allgemeine Beschreibung.....	4
2.2	Updates und Beispielapplikationen.....	4
2.3	Unterschiede zur Version 1 .....	5
3	Rampenprogramme und ihre Möglichkeiten.....	6
3.1	Bestandteile der Rampendefinition .....	6
4	Bedienung.....	7
4.1	Interner Ablauf .....	7
4.2	Die Bedienphilosophie.....	7
4.3	Die Bedienelemente .....	8
4.3.1	Der Drehknopf .....	8
4.3.2	Das Keypad .....	9
4.3.3	Die Event Buttons .....	9
4.4	Die manuelle Bedienebene .....	9
4.5	Die Konfigurationsebene.....	10
5	Steuerung über die serielle Schnittstelle .....	11
5.1	Konzept.....	11
5.2	Die Befehle im Einzelnen .....	12
5.2.1	Kontrolle über den PWM Generator übernehmen .....	12
5.2.2	Kontrolle über den PWM Generator freigeben.....	12
5.2.3	Ausgaben und Rampen ein/ausschalten .....	12
5.2.4	Status der aktuellen Ausgabe .....	13
5.2.5	Bildschirmseite auslesen .....	13
5.2.6	Bildschirmseite setzen.....	13
5.2.7	Duty Cycle für PWM A setzen.....	14
5.2.8	Duty Cycle für PWM B setzen .....	14
5.2.9	Frequenz für PWM A setzen.....	14
5.2.10	Frequenz für PWM B setzen .....	14
5.2.11	Spannung für PWM A setzen.....	14
5.2.12	Spannung für PWM B setzen .....	14
5.2.13	Digitale Ausgänge als Port setzen.....	15
5.2.14	Digitale Ausgänge einzeln setzen .....	15
5.2.15	Digitale Eingänge als Port lesen.....	15
5.2.16	Digitale Eingänge einzeln lesen .....	15
5.2.17	Analoge Ausgangsspannung für alle Kanäle setzen .....	15

5.2.18	Analoge Ausgangsspannung für einen Kanal setzen .....	16
5.2.19	Analoge Eingangsspannung für alle Kanäle lesen .....	16
5.2.20	Analoge Eingangsspannung für einen Kanal lesen .....	16
5.2.21	PWM Werte auslesen.....	16
5.2.22	Initialisierungsbefehl für alle Kanäle .....	17
5.2.23	Firmware Version auslesen .....	17
Anhang A – Anschlussbelegungen und elektrische Spezifikationen .....		18
Anhang B – Details zur Signalerzeugung der PWM Signale .....		21
Anhang C – Dienstprogramme, Beispiele und Updates .....		22
Anhang D - Abbildungen.....		23

## 2 Übersicht

### 2.1 Allgemeine Beschreibung

Der measX PWM Generator gibt - wie schon der Name sagt - pulswertenmodulierten Signale mit einstellbarer Ausgangsspannung und einer maximalen Stromstärke von 2A aus. Dabei ist die Frequenz von 1 Hz bis 5000 Hz in 1 Hz Schritten, der Duty Cycle (Puls Pausenverhältnis) von 0% bis 100% in 0.01% Schritten und die Ausgangsspannung von ca. 0.1V bis ca. 15.0V in 0.1V Schritten einstellbar.

#### *Hinweis:*

*Bitte beachten Sie den Anhang A mit den Anschlussbelegungen und elektrischen Spezifikationen des PWM Generators, um Beschädigungen der einzelnen Komponenten zu vermeiden. Dies gilt insbesondere für die analogen und digitalen Ein- und Ausgänge.*

Der PWM Generator kann über zwei unterschiedlich einstellbare Kanäle externe Netzteile mittels potentialfreier Kontakte ansteuern – für den ersten Kanal steht auch ein eigener Verstärker zur Leistungsabgabe bis 2A zur Verfügung.

Die Einstellung der PWM Werte kann manuell am Gerät oder per Vorgabe über die serielle Schnittstelle erfolgen. Weiter können im Gerät bis zu 20 Rampenprogramme definiert und abgelegt werden, die dann - bei vorgegebener Spannung und Frequenz - Profile über den Duty Cycle abfahren.

Bei der Vorgabe am Gerät können die aktuellen Werte (oder in den Einstellungen der Rampenprogramme die jeweiligen Parameter) sowohl über den Drehknopf an der Front wie auch die Cursortasten neben dem Display eingegeben werden. Die generelle Idee hinter der Bedienung wird in Kapitel 2.1 erläutert – die einzelnen Bedienelemente dann im Anschluss.

Die bis zu 20 Rampenprogramme bestehen jeweils aus einem Hinweg und einem Rückweg, die entweder als Stufentabellen oder als Rampen eingestellt werden können. Dazwischen erfolgen jeweils einstellbare Wartezeiten. Den einzelnen Parametern der Rampen widmet sich Kapitel 3.

Neben den PWM Ausgängen verfügt der PWM Generator über 4 analoge Eingänge, 4 analoge Ausgänge und 8 digitale Ein- und Ausgänge. Die Eingangskanäle werden im Display angezeigt und die Werte der Ausgangskanäle können am PWM Generator vorgegeben werden. Neben der Anzeige/Vorgabe am Gerät selber können die Kanäle auch über die serielle Schnittstelle vorgegeben und gelesen werden.

### 2.2 Updates und Beispielapplikationen

Die Firmware des PWM Generators kann über eine USB Schnittstelle vom Benutzer upgedatet werden - ein entsprechendes Flashprogramm ist im Lieferumfang enthalten.

Eine Beispielapplikation, die alle seriellen Kommandos unterstützt, verdeutlicht die Kommunikation – sie liegt auch als Source Code (Delphi) vor; weitere Beispiele in LabVIEW und DASyLab werden noch erstellt.

Die jeweils neuesten Versionen von Firmware, Dienstprogrammen, Treibern, Handbuch und Beispielen werden von measX online zur Verfügung gestellt – detaillierte Informationen befinden sich in Anhang C.

## 2.3 Unterschiede zur Version 1

In der Grundfunktionalität (Erzeugung der PWM Signale) ist die Auflösung des Duty Cycle nun in Stufen von 0.01% möglich.

Die erreichbare Genauigkeit hängt dabei von der Frequenz ab – die Signalerzeugung des Mikrocontrollers wurde dabei auf eine möglichst genaue Wiedergabe bei den Einer, Zehner, Hunderter und Tausender Frequenzen optimiert (d.h. die Ausgabe mit 100 Hz ist genauer als die Ausgabe mit 101 Hz). Insbesondere die nach Kenntnis von measX häufig verwendeten Frequenzen von 100 Hz, 200 Hz, 300 Hz und 400 Hz werden dabei (vor der Verstärkung) im Rahmen der Timer Genauigkeit (84 MHz) genau wiedergegeben. Natürlich muss dabei der interne Verstärker mit seiner Steigung von 8V/us für die Wahl von Frequenz und Ausgangsspannung berücksichtigt werden.

Die analogen Ausgänge (vier statt einer) sind jetzt von -10 bis 10V einstellbar. Als Option können Module zur galvanischen Trennung (8B Module) der analogen Ausgänge im Gehäuse des PWM Generators durch measX eingesetzt werden.

Die serielle(n) Schnittstelle(n) sind galvanisch entkoppelt – dabei wird zurzeit nur RS232-1 verwendet; die zweite RS232 Schnittstelle ist für spätere Erweiterungen gedacht.

Im Gegensatz zur ersten Version des PWM Generators, der vom Gerät aus die Fernsteuerung aktiv zulassen musste, kann die externe Kontrolle über den PWM Generator nun durch einen speziellen Befehl unabhängig vom Modus erlangt werden – und entsprechend wieder freigegeben werden.

Die analogen Eingänge und digitalen IOs sind komplett neu. Der PWM Generator kann um kundenspezifische Funktionen erweitert werden – so sind z.B. Watchdog Funktionen, Abschaltung auf Pegel oder einfache Regler vorstellbar. Solcherart Zusatzfunktionalität kann von measX als Auftragsarbeit durchgeführt werden.

Die Programmierschnittstelle ist als USB Port herausgeführt.

Die verwendeten Displays sind etwas schneller und haben einige Einstellmöglichkeiten mehr (Helligkeit des Keypads, Töne für Drücken und Loslassen der Tasten des Keypads), die in den entsprechenden Konfigurationsdialogen konfiguriert werden können.

### 3 Rampenprogramme und ihre Möglichkeiten

Im measX PWM Generator können bis zu 20 Rampenprogramme definiert und abgelegt werden, die dann bei vorgegebener Spannung und Frequenz Profile über den Duty Cycle abfahren. Den beiden PWM Ausgängen können Rampenprogramme zugewiesen werden, die dann (zusammen) über den Event Button 2 gestartet werden (per Fernsteuerung über die serielle Schnittstelle geht das auch einzeln).

#### 3.1 Bestandteile der Rampendefinition

1. Nummer der Rampe, unter der diese im Eeprom abgelegt und wieder aufgerufen werden kann. Initialisierung mit Vorgabe Ausgangsspannung (die nur bei PWM A verwendet wird), Frequenz und initialem Duty Cycle sowie einer Haltezeit, bis es mit dem nächsten Schritt weitergeht. Vorgabe eines Wiederholungszählers für die Möglichkeit, die Rampe(n) einstellbar oft zu wiederholen (also ein Sprung zu Schritt 2). Eine unbefristete Wiederholung kann durch Vorgabe von 0 als Wiederholungszähler erreicht werden, ansonsten werden die eingestellten Wiederholungen durchgeführt.

2. Optionaler "Hinweg" mit Start Duty Cycle, End Duty Cycle, Typ und Dauer der Rampe. Als Rampentypen sind eine stetige Veränderung des Duty Cycle von Schwingung zu Schwingung (natürlich im Rahmen der erreichbaren Genauigkeit), sowie zwei treppenartige Rampen möglich.

Im Modus Stufenanzahl kann dabei die Anzahl der Intervalle  $n$  und die Länge jeder Stufe vorgegeben werden. Der Bereich zwischen Start Duty Cycle und End Duty Cycle wird dann in  $n$  gleich große Intervalle aufgeteilt (ergibt  $n+1$  Stufen).

Im Modus Stufenabstand wird dagegen zuerst der Start Duty Cycle für die eingestellte Dauer ausgegeben und dann sukzessive das eingestellte Delta addiert (oder subtrahiert, falls Start Duty Cycle größer als End Duty Cycle ist) solange der berechnete Wert kleiner dem End Duty Cycle ist. Schließlich wird noch der End Duty Cycle für die eingestellte Dauer ausgegeben. Das letzte Intervall kann somit kleiner als die vorherigen Intervalle sein.

Einstellbare Pause nach dem "Hinweg" von 0.01s bis 600.00s. Während der Pause wird der Duty Cycle nicht verändert - egal welcher Abschnitt zuvor ausgeführt wurde.

3. Optionaler "Rückweg" mit Start Duty Cycle, End Duty Cycle, Typ und Dauer der Rampe mit den gleichen Einstellmöglichkeiten wie beim „Hinweg“. Auch wenn die Bezeichnungen "Hinweg" und "Rückweg" suggerieren, dass die erste Rampe aufsteigend und die zweite Rampe abfallend ist, so ist dem nicht so. Beide Rampen können wahlfrei aufsteigend oder abfallend (gleich oder unterschiedlich) sein - das positiv eingegebene Delta im Falle des Modus Stufenabstand wird automatisch addiert oder subtrahiert, um vom Start Duty Cycle zum End Duty Cycle zu kommen.

Einstellbare Pause nach dem "Rückweg" von 0.01s bis 600.00s. Wie auch die Pause nach dem Hinweg ist diese Pause salvatorisch und kann 0.01s bis 600.00s betragen.

Nach Fertigstellung der letzten Wiederholung des Rampenprogramms wird der eingestellte finale Duty Cycle ausgegeben und der PWM Generator springt automatisch wieder in die manuelle Einstellung zurück (wenn das Rampenprogramm von dort gestartet wurde).

## 4 Bedienung

In diesem Kapitel werden das Bedienprinzip und die einzelnen Bedienbildschirme erläutert – beginnend mit einer kurzen Übersicht des internen Ablaufes des PWM Generator Programms und den verschiedenen Zeittakten, die die unterschiedlichen Abläufe anstoßen.

### 4.1 Interner Ablauf

Der measX PWM Generator benutzt für Einstellungen und Generierung der Rohsignale (die durch die Verstärkereinheit auf die nötige Leistung verstärkt werden) einen 32 Bit Mikrocontroller, der in „C++“ programmiert ist.

Dabei wird der Mikrocontroller intern mit 84 MHz getaktet – von diesem Takt werden sowohl die PWM Generierung, wie auch der interne Ablauf abgeleitet. Der interne Ablauf arbeitet mit einer Rate von 100 Hz – in dieser Rate wird die Generierung der PWM Ausgabe (die in Hardware durchgeführt wird) neu konfiguriert (falls geändert), die serielle Schnittstelle für die externe Vorgabe ausgewertet die interne Rampenprogrammgenerierung abgearbeitet sowie die analogen und digitalen Kanäle erfasst und ausgegeben.

Das User Interface mit Bildschirm, Drehknopf und Keypad wird mit 10 Hz aktualisiert bzw. abgefragt. Dabei werden die 10 Hz Aktivitäten auf die „10ms Teilstücke“ des 100 Hz Taktes gleichmäßig aufgeteilt, um „Arbeitsspitzen“ zu bestimmten Zeitpunkten zu vermeiden.

Die Konfiguration der PWM Ausgabe und die Vorgabeart der Werte sind entkoppelt; die Generierung der Signale erfolgt immer gleich – unabhängig von der Art der Vorgabe (manuell, seriell oder Rampenprogramm).

### 4.2 Die Bedienphilosophie

Der PWM Generator hat ein Keypad mit 7 Tasten, bei denen 5 in Form eines Steuerkreuzes mit innenliegender Return Taste angeordnet sind. Dazu gibt es noch einen Drehknopf und zwei Event Buttons zum direkten Auslösen bestimmter Aktionen.

Ziel der Bedienung war eine, trotz der wenigen Tasten, logische Bedienung für manuelle Vorgabe einerseits und Setzen von Optionen andererseits. In der ursprünglichen Version des PWM Generators wurde dazu für eine Reihe von Tasten zwischen der Dauer des Drückens unterschieden – was natürlich gleichzeitig bedeutet, dass diese Aktionen erst beim Loslassen der Tasten durchgeführt werden. Diese (unbeliebte) Mehrfachbelegung wurde soweit wie möglich reduziert. Dabei wurde für einzelne Situationen in Kauf genommen, eine Taste mehr Drücken zu müssen, als das Bedienschema zu durchbrechen.

*Gedankenbild:*

*Die einzelnen Bildschirme sind kleinen Dialogboxen ähnlich, wie sie von Windows, OS X, Linux, etc. bekannt sind. Dort kann man (meist mit der Tab Taste) den Fokus des zu ändernden Controls von einem Control zum nächsten Control bewegen (und durch diverse andere Tasten oder Mausektionen ein Control direkt anwählen). Mittels der Zifferntasten (im Falle von Text oder Zahl Eingaben) oder der Leertaste (zum Auswählen von Optionen) stellt man dann dort die gewünschte Konfiguration her.*

Ähnlich zu diesem Ansatz unterscheidet die Eingabe des PWM Generators zwischen dem Select Modus (zur Auswahl des einzustellenden Controls) und dem Set Modus (zur Veränderung des Wertes dieses Controls). Dabei hat das Steuerkreuz in seinem jeweiligen Modus immer eine sehr ähnliche Funktion für alle Bildschirme.

Während im Set Modus die Steuerkreuz Tasten (und der Drehknopf) den einzustellenden Wert ändern, bewegen sie im Select Modus den Fokus zwischen den Eingabefeldern (Taste Up/Down (siehe Keypad) und Drehknopf) oder zwischen Bildschirmseiten (Taste Right/Left). Der Modus wird mit dem Center Knopf oder dem linken oberen Knopf (Mode) gleich neben dem Display<sup>1</sup> umgeschaltet (außer bei Werten, die nur ein oder ausgeschaltet werden können – da wird der Wert direkt umgeschaltet).

Die graphische Repräsentation ist dabei in der manuellen Bedienebene und der Konfigurationsebene leicht anders – in der manuellen Bedienebene wird das Eingabeelement mit dem Fokus unten rechts auf dem Bildschirm angezeigt; in der Konfigurationsebene blinkt es.

Bleiben nur noch der Aufruf in die Konfigurationsebene und das Verlassen der Konfigurationsebene. Der Aufruf der Konfigurationsebene erfolgt durch das „lange“ Drücken der Center Taste (die Dauer kann in den Einstellungen verändert werden und steht Default auf 750ms). So gelangt man ins Hauptmenü, wo durch die Up/Down Taste oder den Drehknopf der gewünschte Einstelldialog ausgewählt und durch Drücken der Center Taste aufgerufen werden kann.

Schon beim Aufruf des Hauptmenüs hat sich der PWM Generator den Ort des Aufrufs gemerkt und kehrt nach dem Verlassen der Konfigurationsebene an diese Stelle zurück.

In einem Einstelldialog werden die Werte direkt geändert (ein Anlegen zum Verlassen einer Dialogbox in Windows über ESC existiert hier nicht). Der Einstelldialog (der durchaus mehrere Seiten haben kann), wird durch die Back Taste (unten links gleich neben dem Bildschirm) verlassen. Bei kurzem Druck auf die Back Taste geht es zum Hauptmenü zurück und bei langem Druck gleich in die manuelle Bedienebene.

## 4.3 Die Bedienelemente

### 4.3.1 Der Drehknopf

Der Drehknopf ist ein Quadraturencoder, der mittels mit 4 Flankenauswertung ausgewertet wird. Im Select Modus bewegt der Drehknopf den Fokus durch die Eingabefelder (links drehen nach oben/rechts drehen nach unten). Im Set Modus verändert der Drehknopf den einzustellenden Wert um einen kleinen Betrag.

Bei externer Ansteuerung über die serielle Schnittstelle hat der Drehknopf keine Funktion mehr – dann sind nur noch die Left/Right Tasten zum Blättern durch die Anzeigeseiten aktiv.

---

<sup>1</sup> Diese „doppelte“ Ausführung ein- und desselben Befehls ist das Ergebnis von measX internem Feedback, wo denn diese Funktion bedient werden sollte – da haben sich zwei verschiedene Ansätze herausgestellt, die hier beide bedient werden sollen.

#### 4.3.2 Das Keypad

Das Keypad ist Teil des Orbit Displays und ist über eine interne serielle Schnittstelle an den Mikrocontroller angebunden. Darüber werden die Tastatur Befehle eingegeben.

#### 4.3.3 Die Event Buttons

Die beiden Event Buttons sind in der Standard Version des PWM Generators mit festen Funktionen belegt.

Der obere Button (EB1) schaltet im manuellen Modus die Ausgabe des ersten PWM Kanals ein und aus – so können die Werte im ausgeschalteten Zustand verändert werden und dann die Ausgabe gestartet werden.

Bei eingeschalteten PWM Ausgängen leuchtet der Button (grün), bei ausgeschalteten PWM Ausgängen nicht.

Im Fernsteuermodus (über die serielle Schnittstelle) hat der Button keine Funktion beim Betätigen – die interne LED des Buttons leuchtet wie im manuellen Modus, wenn die Ausgabe eingeschaltet ist und zeigt auch ein Ein- oder Ausschalten über die serielle Schnittstelle an.

Der untere Button (EB2) startet im manuellen Modus die Ausgabe vordefinierter und dem ersten PWM Kanal zugewiesenen Rampenprogrammen. Ist mindestens eine Rampe definiert und zugewiesen, dann leuchtet nach Betätigung der Button ebenfalls grün, bis die Rampe wieder ausgeschaltet wird (durch erneute Betätigung) oder zu Ende abgearbeitet wurde.

Im Fernsteuermodus (über die serielle Schnittstelle) hat auch dieser Button keine Funktion beim Betätigen – die interne LED des Buttons leuchtet aber, wenn mindestens eine Rampe ausgegeben wird. Rampen auf dem zweiten PWM Ausgang können ausschließlich über die serielle Schnittstelle gestartet und angehalten werden.

Wenn im Fernsteuermodus beide Knöpfe für mindestens 5s gleichzeitig gedrückt werden, wird die Bedienkontrolle wieder an das Gerät selbst zurückgegeben. Dies kann allerdings zu unerwünschten Effekten im fernsteuernden Programm führen und ist nur als eine Art Rettungsanker, falls das fernsteuernde Programm in der aktiven Fernsteuerung ausfällt und die Kontrolle nicht ordnungsgemäß zurück an den PWM Generator übertragen kann. Ein Aus- und wieder Einschalten des PWM Generators (natürlich mit ein paar Sekunden Pause dazwischen) erfüllt aber den gleichen Zweck, da der Fernsteuerzustand nicht im Eeprom abgelegt wird.

#### 4.4 Die manuelle Bedienebene

---

*Wird ergänzt*

---



## 4.5 Die Konfigurationsebene

---

*Wird ergänzt*

---

## 5 Steuerung über die serielle Schnittstelle

### 5.1 Konzept

Die Kommandostruktur ist angelehnt (bzw. gleich) zur Struktur der alten PWM Generatoren und ist auf der Seite vom PC zum PWM Generator komplett in Text realisiert (Antworten des PWM Generators an den PC können auch binär sein).

Ein Befehl startet immer mit einem STX (#2) und endet immer mit einem ETX (#3).

Der PWM Generator empfängt die Daten der seriellen Schnittstelle auf Port 1 des PWM Generators und wartet auf ein STX Zeichen. Dann liest er die Daten bis zum ETX Zeichen in einen Puffer und dekodiert diesen Puffer. Es gibt Befehle ohne Parameter, mit einem Parameter und mit mehreren Parametern. Ein Befehl darf max. 64 Zeichen lang sein (inkl. STX und ETX) – ansonsten wird er vom PWM Generator verworfen. Bei Befehlen mit mehreren Parametern darf jeder einzelne Parameter max. 16 Zeichen lang sein – ansonsten wird der komplette Befehl verworfen. Diese Grenzen stellen in den bisher implementierten und geplanten Befehlen allerdings keine Beschränkung dar (nicht mal theoretisch) und dienen der Fehlererkennung eines möglicherweise inkorrekt gesendeten Befehls.

Der PWM Generator schaut im 10ms Takt auf die Schnittstelle und wertet ankommende Befehle aus. Dabei ist die Dauer der Auswertung und Ausführung (genauer gesagt der Start der letzten Auswertung im jeweiligen 10ms Zyklus) auf 0.33ms beschränkt – weitere Befehle werden im nächsten 10ms Zyklus abgearbeitet. Die Schnittstelle wird mit 115200 Baud betrieben, 8 Datenbits, 1 Stoppbit und NOPARITY (das sind die de facto Standard Einstellungen). Zum Anschluss ist ein Nullmodemkabel notwendig (Pin 2 und Pin 3 „überkreuz“ und nicht direkt verbunden).

## 5.2 Die Befehle im Einzelnen

Wie schon weiter oben beschrieben, werden alle Befehle von STX und ETX „eingerahmt“ – diese werden in den weiteren Befehlsbeschreibungen nicht jedes Mal explizit aufgeführt.

### 5.2.1 Kontrolle über den PWM Generator übernehmen

Kommando: X (keine Parameter)

Kommentar: Setzt den PWM Generator auf externe Kontrolle, so dass er weitere Befehle über die RS232 Schnittstelle annimmt. Dabei werden laufende Rampen nicht unterbrochen (dafür gibt es einen separaten Befehl) und auch der Status der Ausgabe (die über den Event Button 1 für alle Kanäle ein- und ausgeschaltet werden kann) wird nicht verändert. Die LEDs in den Event Buttons bleiben aus, wenn sie im manuellen Betrieb aus waren. Im Display des PWM Generators wird angezeigt, ob die Kontrolle manuell oder über die serielle Fernsteuerung erfolgt.

### 5.2.2 Kontrolle über den PWM Generator freigeben

Kommando: x (keine Parameter)

Kommentar: Setzt den PWM Generator auf interne Kontrolle. Alle weiteren Befehle über die RS232 Schnittstelle werden verworfen bis zum nächsten „Kontrolle übernehmen“. Dabei werden allerdings alle Rampenprogramme angehalten und die Ausgabe des PWM Generators ausgeschaltet (dann sind beide Event Buttons aus).

### 5.2.3 Ausgaben und Rampen ein/ausschalten

Kommando: M

Parameter: Ziffer mit dem Befehl:

- 0: Nummer der manuell konfigurierten Rampe für PWM A setzen (1..20)  
Beispiel: M018 (eingepackt in STX ETX) setzt die 18te Rampe für PWM A.
- 1: Ausgabe auf PWM A einschalten
- 2: Aktuelle Rampe/Stufenprogramm auf PWM A starten
- 3: Ausgabe auf PWM A ausschalten (DC = 0, V = 0 solange gestoppt ist)
- 4: Aktuelle Rampe/Stufenprogramm auf PWM A beenden
- 5: Nummer der manuell konfigurierten Rampe für PWM B setzen (1..20)  
Beispiel: M117 (eingepackt in STX ETX) setzt die 17te Rampe für PWM B.
- 6: Ausgabe auf PWM B einschalten (bei Geräten mit zweikanaligem Verstärker)
- 7: Aktuelle Rampe/Stufenprogramm auf PWM B starten
- 8: Ausgabe auf PWM B ausschalten (bei Geräten mit zweikanaligem Verstärker)
- 9: Aktuelle Rampe/Stufenprogramm auf PWM B stoppen (DC = 0)
- A: Bei beiden PWM Kanälen Ausgabe und Rampenprogramm starten
- B: Bei beiden PWM Kanälen Ausgabe und Rampenprogramm stoppen

Kommentar: Startet oder stoppt die Ausgabe und/oder die Rampengenerierung einzeln oder zusammen.

#### 5.2.4 Status der aktuellen Ausgabe

Kommando: m

Parameter: Ziffer mit der Nachfragefunktion:

- 0: Ausgabestatus PWM A  
Binäre Rückgabe: 0 oder 1 als 8 Bit Integer (0 – Aus, 1 – An)
- 1: Ausgabestatus PWM B (bei Geräten mit zwei Verstärkern)  
Binäre Rückgabe: 0 oder 1 als 8 Bit Integer (0 – Aus, 1 – An)
- 2: Nummer der aktuellen Rampe PWM A  
Binäre Rückgabe: 0..20 als 8 Bit Integer (0: keine Rampe zugewiesen)
- 3: Nummer der aktuellen Rampe PWM B  
Binäre Rückgabe: 0..20 als 8 Bit Integer (0: keine Rampe zugewiesen)
- 4: Status Rampe PWM A  
Binäre Rückgabe: Gesamtlaufzeit in 0.01s als 32Bit Integer (0 – Aus)  
Aktuelle Laufzeit in 0.01s als 32Bit Integer
- 5: Status Rampe PWM B  
Binäre Rückgabe: Gesamtlaufzeit in 0.01s als 32Bit Integer (0 – Aus)  
Aktuelle Laufzeit in 0.01s als 32Bit Integer
- 6: Gesamt Status Ausgabe und Rampenprogramme  
Binäre Rückgabe: 4 8-Bit Integer und 4 32-Bit Integer in obiger Reihenfolge

Kommentar: Fragt den Status der Ausgabe des PWM Generators ab – möglicherweise hilfreich nach der Übernahme der Kontrolle, um laufende Rampen noch beenden zu können oder im ferngesteuerten Betrieb anzuzeigen, wie lange die Rampen schon bzw. noch laufen.

#### 5.2.5 Bildschirmseite auslesen

Kommando: s (keine Parameter)

Rückgabe: Bildschirmseite als Byte

#### 5.2.6 Bildschirmseite setzen

Kommando: S

Parameter: Nummer der Bildschirmseite (1..5)

Kommentar: Beim Übernehmen der externen Kontrolle über die serielle Schnittstelle wird (auch wenn gerade Eingabemenu geöffnet ist) auf eine der direkten Anzeigeseiten geschaltet. Sollten weitere Anzeigeseiten hinzukommen erhöht sich der mögliche Parameter hier.

#### 5.2.7 Duty Cycle für PWM A setzen

Kommando: D

Parameter: Duty Cycle in 1/100 Prozent (0..10000)

Kommentar: Änderungen des Duty Cycle erfolgen ohne Unterbrechung der Ausgabe und werden komplett in Hardware ausgeführt.

#### 5.2.8 Duty Cycle für PWM B setzen

Kommando: d

Parameter: Duty Cycle in 1/100 Prozent (0..10000)

Kommentar: Änderungen des Duty Cycle erfolgen ohne Unterbrechung der Ausgabe und werden komplett in Hardware ausgeführt.

#### 5.2.9 Frequenz für PWM A setzen

Kommando: F

Parameter: Frequenz in Hz (1..5000)

Kommentar: Änderungen der Frequenz können zu kurzfristigen Unterbrechungen (im  $\mu$ s Bereich) der Ausgabe führen, da die Ausgabe an bestimmten Frequenzgrenzen angehalten und gleich wieder gestartet werden muss. Die Frequenzgrenzen, an denen die Signalerzeugung kurzfristig unterbrochen werden muss, sind in Anhang X beschrieben.

#### 5.2.10 Frequenz für PWM B setzen

Kommando: f

Parameter: Frequenz in Hz (1..5000)

Kommentar: Änderungen der Frequenz können zu kurzfristigen Unterbrechungen (im  $\mu$ s Bereich) der Ausgabe führen, da die Ausgabe an bestimmten Frequenzgrenzen angehalten und gleich wieder gestartet werden muss.

#### 5.2.11 Spannung für PWM A setzen

Kommando: V

Parameter: Spannung in 1/10 Volt (1..150)

#### 5.2.12 Spannung für PWM B setzen

Kommando: v

Parameter: Spannung in 1/10 Volt (1..150)

Kommentar: Im Falle eines PWM Generators mit einkanaligem internem Verstärker, hat der Befehl keine Wirkung (sprich: er ist auch nicht schädlich).

#### 5.2.13 Digitale Ausgänge als Port setzen

Kommando: P

Parameter: Port als Byte (0..255)

Kommentar: DO1 ist das niederwertigste Bit, DO8 ist das höchstwertige Bit des Ports. Ein Wert von 42 setzt also DO6, DO4 und DO2.

#### 5.2.14 Digitale Ausgänge einzeln setzen

Kommando: p

Parameter: 2 Ziffern mit folgender Bedeutung:  
Ziffer 1: Nummer des digitalen Ausgangs (1..8)  
Ziffer 2: Zu setzender Status (0 oder 1)

Kommentar: Bei ungültigen Werten für die Nummer des Ausgangs oder den zu setzenden Wert wird der Befehl ignoriert.

#### 5.2.15 Digitale Eingänge als Port lesen

Kommando: P

Rückgabe: Wert der 8 digitalen Eingänge als Port (bitcodiertes Byte).

Kommentar: DI1 ist das niederwertigste Bit, DI8 ist das höchstwertige Bit des Ports. Ein Wert von 42 bedeutet also, dass DI6, DI4 und DI2.

#### 5.2.16 Digitale Eingänge einzeln lesen

Kommando: p

Parameter: Nummer des digitalen Eingangs (1..8)

Rückgabe: Byte: 0 – Eingang nicht gesetzt; 1 – Eingang gesetzt

Kommentar: Bei ungültigen Werten für die Nummer des Eingangs wird der Befehl ignoriert.

#### 5.2.17 Analoge Ausgangsspannung für alle Kanäle setzen

Kommando: A

Parameter: Die 4 auszugebenden Werte als Text durch Semikolon getrennt (mit einem Punkt als Dezimaltrennzeichen).

Kommentar: Der Wertebereich ist -10.5V bis 10.5V – mehr als 3 Nachkommastellen sind unnötig. Am Ende der Eingabe darf kein Semikolon stehen. Werte außerhalb des zulässigen Bereiches werden ignoriert. Zurzeit wird eine unvollständige Parameterliste (also weniger als 4 Kanäle) nicht ignoriert, sondern es werden die ersten gültigen Kanäle genommen. Dies muss in späteren Versionen aber nicht so bleiben.

#### 5.2.18 Analoge Ausgangsspannung für einen Kanal setzen

Kommando: a

Parameter: Ziffer für die Kanalnummer (1..4) direkt gefolgt von dem auszugebenden Wert als Text (mit einem Punkt als Dezimaltrennzeichen).

Kommentar: Der Wertebereich ist -10.5V bis 10.5V – mehr als 3 Nachkommastellen sind unnötig. Beispiel a33.141 setzt AO3 auf ca. Pi.

#### 5.2.19 Analoge Eingangsspannung für alle Kanäle lesen

Kommando: R (keine Parameter)

Rückgabe: Alle 4 Eingangsspannungen als float Werte einfacher Genauigkeit (also 4 mal 4 Byte)

Kommentar: Der Messbereich ist ca. 0V bis 10.5V (je nach Kalibrierung). Obwohl unipolar misst der A/D Wandler wenige mV in den negativen Bereich.

#### 5.2.20 Analoge Eingangsspannung für einen Kanal lesen

Kommando: r

Parameter: Ziffer für die Kanalnummer (1..4).

Rückgabe: Eingangsspannungen als float Wert einfacher Genauigkeit (4 Byte)

Kommentar: Der Messbereich ist ca. 0V bis 10.5V (je nach Kalibrierung). Obwohl unipolar misst der A/D Wandler wenige mV in den negativen Bereich.

#### 5.2.21 PWM Werte auslesen

Kommando: q

Parameter: Ziffer mit der Nachfragefunktion:

- 1: PWM A Frequenz in Hz  
Binäre Rückgabe: (1..5000) als 16 Bit Integer
- 2: PWM A Duty Cycle in 0.01%  
Binäre Rückgabe: (0..10000) als 16 Bit Integer
- 3: PWM A Spannung in 0.1V  
Binäre Rückgabe: (0..150) als 16 Bit Integer
- 4: PWM B Frequenz in Hz  
Binäre Rückgabe: (1..5000) als 16 Bit Integer
- 5: PWM B Duty Cycle in 0.01%  
Binäre Rückgabe: (0..10000) als 16 Bit Integer
- 6: PWM B Spannung in 0.1V  
Binäre Rückgabe: (0..150) als 16 Bit Integer
- 7: PWM A und B Duty Cycle in 0.01%  
Binäre Rückgabe: 2 16 Bit Integer (0..10000)
- 8: PWM A Frq, DC, U, PWM B Frq, DC, U  
Binäre Rückgabe: 6 16Bit Integer mit Werten wie oben

Kommentar: Fragt einzelne Parameter der PWM Vorgabe ab – bzw. beide Duty Cycle bzw. alle Parameter en bloc. Im Falle eines PWM Generators mit einkanaligem internem Verstärker, sind die zurückgegebenen Spannungswerte für PWM B zufällig.

#### 5.2.22 Initialisierungsbefehl für alle Kanäle

Kommando: I (keine Parameter)

Rückgabe: -

Kommentar: Setzt beide PWM Kanäle auf 0 V, 100 Hz und 50% DC. Setzt alle analogen Ausgänge auf 0V und alle digitalen Ausgänge auf 0.

#### 5.2.23 Firmware Version auslesen

Kommando: i (keine Parameter)

Rückgabe: Gibt die Firmware Version als Text zurück

## Anhang A – Anschlussbelegungen und elektrische Spezifikationen

Die Frontseite des PWM Generators enthält die Anschlüsse für die PWM Ausgänge und die Bedienelemente (Ein/Aus Taster, Keypad, Bildschirm, Event Buttons). Die Rückseite des PWM Generators enthält die analogen und digitalen Ein und Ausgänge, sowie die seriellen Schnittstellen, die Programmierschnittstelle und den Netzanschluss.

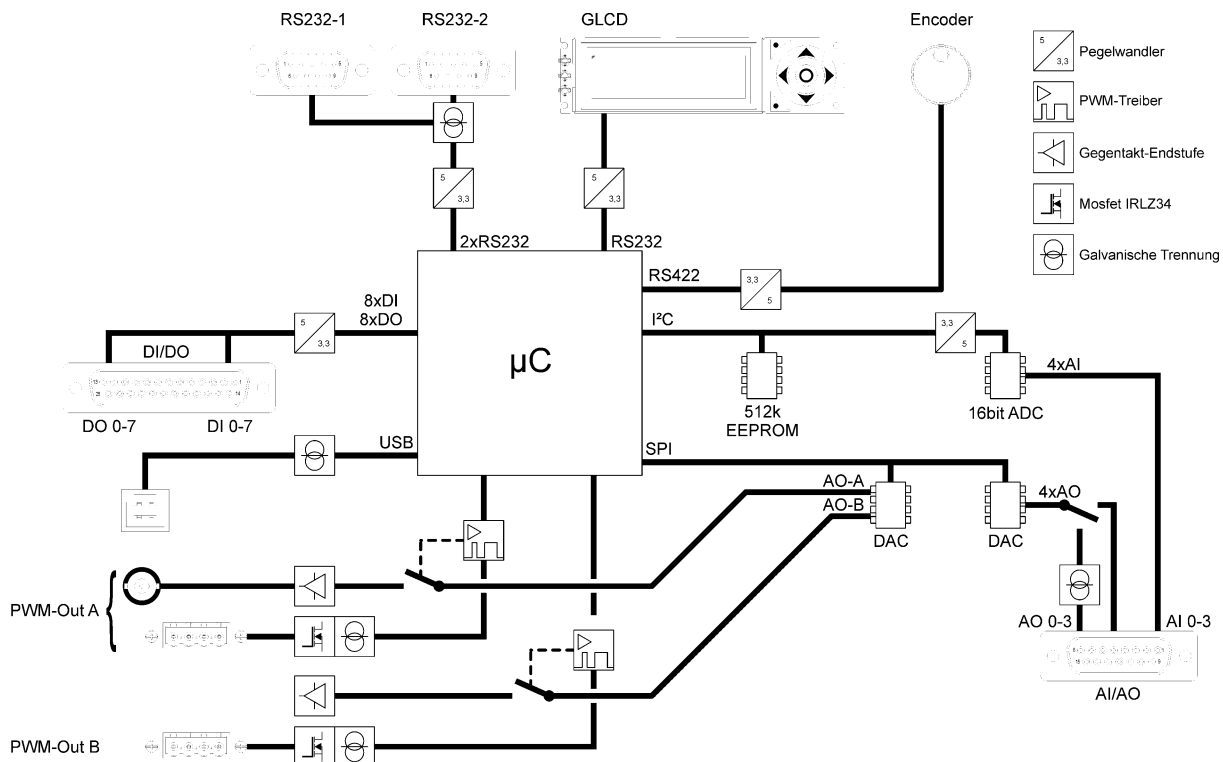
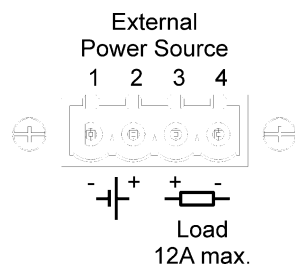


Abbildung 1: Blockschaltbild

Die Belegungen der Steckverbindungen im Einzelnen:

### 1. PWM Ausgang (Frontseite)

PWM Ausgang - extern	
Pin	Funktion
1	Quelle
2	Quelle
3	Prüfling
4	Prüfling



### PWM-Out A/B

External Power Source  
Schaltstufe: MOSFET mit open drain  
Schaltspannung: max. 35VDC  
Schaltstrom: max. 12A

Abbildung 2: PWM externe Quelle

**Anmerkung:**

Internal  
Power Source



Load  
2A max.

Abbildung 3: PWM  
Interner Verstärker

Der PWM Ausgang A mit eigenem Verstärker wird über eine BNC Buchse ausgegeben. Bei PWM Ausgang A darf *entweder* nur der interne Verstärker (BNC) *oder* die externe Quelle belegt sein.

**PWM-Out A:**

Internal Power Source - Schaltstufe: Gegentaktendstufe mit:

Uout: 0V bis +15.0V

I<sub>max</sub>: 2A

## 2. Analoge Ein- und Ausgänge (Rückseite)

AI/AO DSub15	
Pin	Funktion
1	AI 1
2	AI 2
3	AI 3
4	AI 4
5	AO 1
6	AO 2
7	AO 3
8	AO 4
9	GND
10	GND
11	GND
12	AOGND (1)*
13	AOGND (2)*
14	AOGND (3)*
15	AOGND (4)*

**AI/AO**



Abbildung 4: Pinbelegung AI/AO

Alle analogen-Ein- und Ausgänge sind massebezogen und nicht entkoppelt. Um Beschädigungen zu vermeiden, sollten die Analogeingänge bei ausgeschaltetem PWM Generator spannungslos sein.

\*) Im Falle der Option „Galvanisch getrennte AO“ durch 8B Module.

**Analoge Eingänge**

Eingangsmessbereich: 0V bis +10V

Eingangsimpedanz: 270 KOhm

**Analoge Ausgänge**

Ausgangsspannungsbereich: -10V bis +10V

Ausgangsimpedanz: ca. 0.5 Ohm

Ausgangsstrom: max. 5mA (Last >2 KOhm)

### 3. RS 232 Ports

RS232 DSub9	
Pin	Funktion
1	
2	RX
3	TX
4	
5	GND
6	
7	
8	
9	

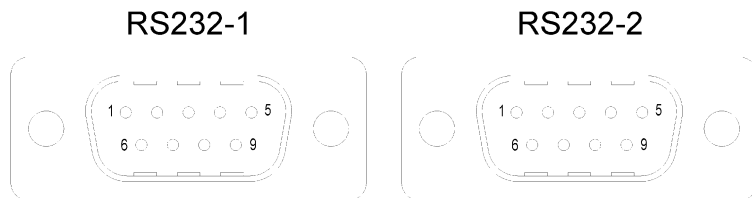


Abbildung 5: Pinbelegung serielle Schnittstelle

#### Hinweis:

Das Verbindungskabel zum PC muss ein sog. „Nullmodem Kabel“ sein; sprich: RX und TX werden nicht durchverbunden, sondern gekreuzt.

### 4. Digitale Ein- und Ausgänge

DI/DO DSub25	
Pin	Funktion
1	DI 1
2	DI 3
3	DI 5
4	DI 7
5	GND
6	GND
7	GND
8	GND
9	GND
10	DO 2
11	DO 4
12	DO 6
13	DO 8
14	DI 2
15	DI 4
16	DI 6
17	DI 8
18	GND
19	GND
20	GND
21	GND
22	DO 1
23	DO 3
24	DO 5
25	DO 7

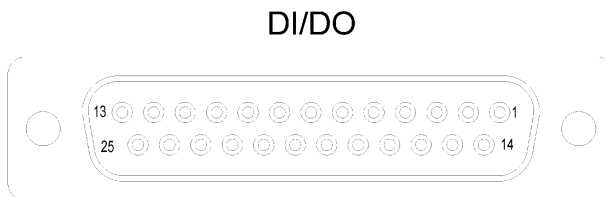


Abbildung 6: Pinbelegung DI/DO

Alle digitalen Ein- und Ausgänge sind nicht entkoppelt.

Um Beschädigungen zu vermeiden, sollten die Digitaleingänge bei ausgeschaltetem PWM Generator spannungslos sein.

#### Digitalinput

LOW-Level: 0V bis +0,8V  
HIGH-Level: +2,4V bis 5,0V  
Absolute Grenzwerte: -0,5V bis +5,5V

#### Digitaloutput

LOW-Level: +0,1V bis +0,6V (je nach Belastung)  
HIGH-Level: +2,7V bis 4,5V (je nach Belastung)  
Ausgangsstrom:  
I sourcing: max. 20mA  
I sinking: max. 20mA

## Anhang B – Details zur Signalerzeugung der PWM Signale

Die folgenden Informationen sind zur Bedienung und Benutzung nicht notwendig, mögen aber für das Verständnis der Genauigkeit der PWM Ausgabe interessant sein. Hier wird nur die Erzeugung des Rohsignals vom Mikrocontroller beleuchtet.

Der verwendete Mikrocontroller ist ein Cortex M3 (Atmel SAM3XE8). Dieser stellt eine Reihe von Hardware PWM Funktionen zur Verfügung, von denen zwei für den measX PWM Generator verwendet werden.

Jede PWM Zelle des Mikrocontrollers besteht (stark vereinfacht) aus zwei 16 Bit Zählern von denen einer die Frequenz und der andere die Dauer des High Anteils steuert. Diese beiden Zähler erhalten einen Eingangstakt, der in zwei Stufen vom Prozessortakt heruntergeteilt wird. Jede PWM Zelle hat zwei Sätze von Registern zum Ändern dieser beiden Zähler – einmal zum initialen Setzen (was bei stehender PWM Ausgabe verwendet wird) und zum anderen zum Updaten während der Ausgabe. Im zweiten Fall puffert die PWM Zelle diese beide neuen Werte, führt das „alte“ PWM Signal bis zum Ende einer Periode durch, schaltet auf die neuen Werte (in Hardware) um und macht mit diesen gleich die nächste Periode ohne Unterbrechung.

Die Taktung der PWM Zelle wird vom Prozessortakt abgeleitet (84MHz) und kann in einer ersten Stufe um eine Zweierpotenz ( $2^0$  bis  $2^{10}$ ) heruntergeteilt werden und in einer zweiten Stufe um einen Wert zwischen 1 und 255. Dafür gibt es genau zwei unabhängige Pfade (deswegen auch 2 Kanäle im measX PWM Generator). Bei Änderung der Taktrate der PWM Zelle, muss diese vorher angehalten werden, die Taktrate geändert werden und dann neu gestartet werden. Das Anhalten geschieht dabei über ein Signal, dass angehalten werden soll und dem Abfragen eines Status, ob denn die Periode zu Ende ist (bei Frequenzen bis 1 Hz kann man darauf nicht warten, sondern muss das in einem Hintergrundprozess erledigen – in Konkurrenz mit vielen anderen Dingen wie Senden der gepufferten Daten zu diversen Schnittstellen, etc.). Dieser Hintergrundprozess kommt sehr häufig zur Ausführung (zurzeit  $> 10^5$  mal in der Sekunde) und die zurzeit längste Aktion ist etwa 100us „lang“. Dies ist damit dann auch die längste zu erwartende Pause zwischen zwei Perioden eines PWM Zyklus in denen die Frequenz so stark geändert wurde, dass eine neue Taktrate für die PWM Zelle nötig war.

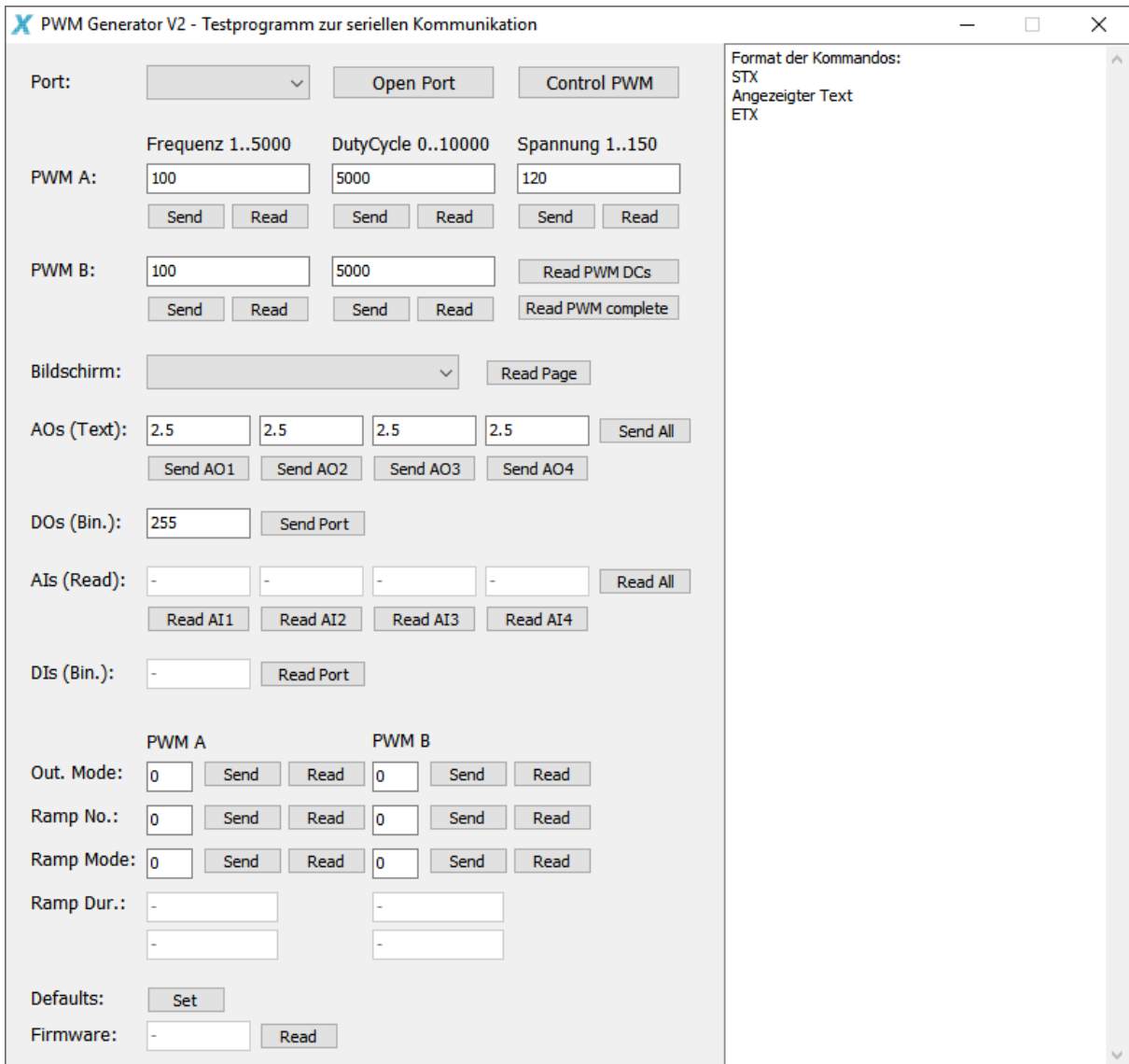
Der nötige Takt einer PWM Zelle ergibt sich aus der gewünschten Frequenz und der gewünschten Genauigkeit des Duty Cycle sowie der „Breite“ der Register der PWM Zelle. Diese „Breite“ ist durch den verwendeten Mikrocontroller fest vorgegeben (16 Bit – also 0 bis 65536 als Registerwerte) – und die gewünschte Auflösung des Duty Cycle (0.01%). Damit würde eine optimale Taktrate für eine Frequenz 10000 (bzw. das höchste Vielfache davon im 16Bit Bereich) für eine Periode betragen.

Das geht nun nicht für jede Frequenz – deswegen wurden die Einer, Zehner, Hunderter und Tausender Frequenzen (also 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 2000, 3000, 4000, 5000 Hz) möglichst optimale Teiler bestimmt.

Beim Wechsel auf bzw. über so eine „Frequenzgrenze“ kommt es typischerweise zu einem Anhalten/Wiederstarten – bei Veränderungen des Duty Cycle oder der Ausgabespannung nie.

## Anhang C – Dienstprogramme, Beispiele und Updates

Mit dem Testprogramm zur seriellen Kommunikation können Sie die Befehle austesten – bzw. den Source Code als Referenz zur Programmierung verwenden.



The screenshot shows the 'PWM Generator V2 - Testprogramm zur seriellen Kommunikation' window. It features a left sidebar with various control sections and a right sidebar for command format settings.

**Left Sidebar Controls:**

- Port:** A dropdown menu, 'Open Port', and 'Control PWM' buttons.
- PWM A:** Fields for 'Frequenz 1..5000' (value: 100), 'DutyCycle 0..10000' (value: 5000), and 'Spannung 1..150' (value: 120). Each has 'Send' and 'Read' buttons.
- PWM B:** Fields for 'Frequenz 1..5000' (value: 100) and 'DutyCycle 0..10000' (value: 5000). It includes 'Send' and 'Read' buttons, a 'Read PWM DCs' button, and a 'Read PWM complete' button.
- Bildschirm:** A dropdown menu and a 'Read Page' button.
- AOs (Text):** Four input fields (all with '2.5') and a 'Send All' button. Below are 'Send AO1', 'Send AO2', 'Send AO3', and 'Send AO4' buttons.
- DOs (Bin.):** An input field (value: 255) and a 'Send Port' button.
- AIIs (Read):** Four input fields (all with '-') and a 'Read All' button. Below are 'Read AI1', 'Read AI2', 'Read AI3', and 'Read AI4' buttons.
- DIIs (Bin.):** An input field (value: '-') and a 'Read Port' button.
- PWM A and PWM B Section:**
  - Out. Mode:** Input fields (both '0') with 'Send' and 'Read' buttons.
  - Ramp No.:** Input fields (both '0') with 'Send' and 'Read' buttons.
  - Ramp Mode:** Input fields (both '0') with 'Send' and 'Read' buttons.
  - Ramp Dur.:** Two input fields (both '-') for each PWM channel.
- Defaults:** A 'Set' button.
- Firmware:** An input field (value: '-') and a 'Read' button.

**Right Sidebar:** 'Format der Kommandos:' section with a list: STX, Angezeigter Text, ETX.

Die PWM-Spannungen werden dabei in 1/10 Volt eingegeben, die Frequenzen als ganzzahlige Werte und die Duty Cycles als 1/100 Prozent.

Für das Ein- und Ausschalten der Ausgänge und der Rampen werden „1“ für das Einschalten bzw. „0“ für das Ausschalten eingegeben – die Rückantworten sind entsprechend kodiert und werden in dieselben Felder eingetragen. Die Gesamt- und Restlaufzeit bei der Abfrage der Rampenstatus sind in der Auflösung 1/100 Sekunde angegeben (dies ist die interne Taktzeit für Abläufe des PWM Generators.)

## Anhang D - Abbildungen

Abbildung 1: Blockschaltbild .....	18
Abbildung 2: PWM externe Quelle.....	18
Abbildung 3: PWM Interner Verstärker .....	19
Abbildung 4: Pinbelegung AI/AO.....	19
Abbildung 5: Pinbelegung serielle Schnittstelle .....	20
Abbildung 6: Pinbelegung DI/DO .....	20