

# Bedienungsanleitung

## Uhrtaktgeber Taktgeberkern

Vertraulichkeitsgrad:	Streng öffentlich
Herausgebende Abteilung:	FREMO-Dienststelle Mascherode

Freigegeben	BWis	FREMO_BS		gez. BWis
Geprüft				gez.
Erstellt	BWis	FREMO_BS	17.05.2007	gez.BWis
	Name	Org.-Einheit, Tel.	Datum	Unterschrift

Bernd Wisotzki  
Heinrich-Netzel-Weg 5  
D 38126 Braunschweig, Mascherode

Telefon +49 531 310789-4  
Telefax +49 531 214789-90

E-mail: [wsb56@sourceforge.net](mailto:wsb56@sourceforge.net)  
Web: [www.modellbahn.wisotzki.org](http://www.modellbahn.wisotzki.org)  
Web: [www.ba.fremo-web.org](http://www.ba.fremo-web.org)

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemeines</b> .....	<b>5</b>
1.1	Bedienung im Überblick.....	5
1.2	Das LCD .....	5
1.2.1	Zeile 1 .....	5
1.2.2	Zeile 2 .....	5
1.3	Die Tasten.....	6
1.4	Die Hauptmenüs .....	6
1.4.1	Init (Initialisierung).....	6
1.4.2	Norml (Normalbetrieb) .....	6
1.4.3	Progr (Programmierung).....	6
1.4.4	Shutd (Shutdown) .....	7
1.5	Weiterer Aufbau der Anleitung.....	7
<b>2</b>	<b>Init</b> .....	<b>8</b>
2.1	„Begrüßung“ .....	8
2.2	Init (Systemstatus) .....	8
<b>3</b>	<b>Norml (Normalbetrieb)</b> .....	<b>9</b>
3.1	Einstieg .....	9
3.2	NmUhr (Normaler Uhrbetrieb) .....	9
3.3	NmSvr (Schneller Vorlauf bis ...).....	9
3.4	NmVlz (Zeitverkürzung) .....	10
3.5	NmSZt (Startzeit) .....	10
3.5.1	Minte .....	11
3.5.2	Stnde.....	11
3.6	Kleine Amerkung.....	11
<b>4</b>	<b>Progr (Programmierung)</b> .....	<b>12</b>
4.1	Einstieg .....	12
4.2	PrPuB.....	12
4.3	PrSe0 .....	12
<b>5</b>	<b>Shutd (Shutdown)</b> .....	<b>13</b>
5.1	Einstieg .....	13
5.2	Ende.....	13
<b>6</b>	<b>Hintergründiges</b> .....	<b>14</b>
6.1	Warum „Shutdown“? .....	14
6.2	Was bringen spätere Versionen? .....	14
6.3	Wo finde ich weiter Informationen? .....	14
<b>7</b>	<b>Änderungsindex</b> .....	<b>15</b>



# 1 Allgemeines

Dieses Dokument soll keine voll erschöpfende Bedienungsanleitung sein, sondern eine Übersicht über die Menüs und ihre Funktionen. Wer Ergänzungen für notwendig erachtet, schreibe sie bitte selbst und lasse sie mir dann zukommen.

## Eine große Bitte:

Erste die Anleitung lesen und dann posten und fragen!

## 1.1 Bedienung im Überblick

Bei der Entwicklung habe ich versucht eine möglichst einfache und durchgängige Bedienung hinzubekommen und dabei trotzdem die vorhandenen Ressourcen zu schonen, sowohl Hardware und Software als auch den Geldbeutel.

Somit ist die Bedienung auf ein einfaches LCD mit 2\*16 Zeichen und vier Tasten reduziert, ähnlich der Bedienung der Menüs bei Mobil- oder sonstigen Telefonen.

## 1.2 Das LCD

Grundsätzlich ist die Anzeige (fast) immer gleich aufgebaut. Die Ausnahmen werden explizit beschrieben:

```
12:34 10,2 Norml  
21:56      NmUhr
```

So wird jeder Menüpunkt hier dargestellt, und dieser allgemeine Aufbau wird bei den einzelnen Menüpunkten nicht mehr beschrieben, nur die Ausnahmen.

### 1.2.1 Zeile 1

- 1-5: Startzeit (da fängt „morgens“ der Fahrplan an)
- 6: Leerzeichen
- 7-10: Zeitverkürzung (1:1,0 bis 1:11,9)
- 11: Leerzeichen
- 12-16: Aktuelles Menü (da bin ich gerade)

### 1.2.2 Zeile 2

- 1-5: aktuelle Zeit

- 6-11: Leerzeichen
- 12—16: Zielmenü oder Zielaktion (das will ich im nächsten Schritt tun)

## 1.3 Die Tasten

Wie schon gesagt, beschränken sich die Eingabemöglichkeiten auf vier Tasten, auch sie werden nur hier beschrieben:

- Enter (Eingabe): Führt einen Menüpunkt aus, speichert Daten etc.
- Zurück: Notausgang, zurück ohne Speichern, etc.
  
- Up: Schaltet einen Menüpunkt weiter, oder zählt hoch (auch Dauerfeuer)
- Down: Schaltet einen Menüpunkt zurück oder zählt runter...

Ich habe versucht, nur sinnvolle Tastenkombinationen zuzulassen, hoffentlich ist es überall gelungen.

## 1.4 Die Hauptmenüs

Im Moment haben wir vier Hauptmenüs, wovon zwei sehr kurz sind. Das kann sich in späteren Versionen durchaus ändern.

### 1.4.1 Init (Initialisierung)

Hier agiert im Wesentlichen der Mikrocontroller. Er holt sich den letzten Stand aus dem EEPROM, wertet ihn aus, zeigt das Ergebnis an und gibt die Möglichkeiten nach „Norml“ oder „Progr“ zu gehen. Die Vorbelegung hängt vom Ergebnis der Auswertung ab.

### 1.4.2 Norml (Normalbetrieb)

Das ist der Normalbetrieb des Uhrtaktgebers. Hier sage ich „Uhr läuft“, „Uhr steht“, „schneller Vorlauf bis...“ oder „Zeitverkürzung...“. Ausserdem kann ich natürlich nach „Progr“ oder „Shtud“ wechseln.

### 1.4.3 Progr (Programmierung)

Hier kann ich verschiedene Dinge einprogrammieren. Derzeit eigentlich nur das Setzen auf 00:00 Uhr und in sehr engen Grenzen die Pulsdauer.

## 1.4.4 Shutd (Shutdown)

Unscheinbar, aber wichtig: Der „Shutdown“. Denn irgendwie muss ich dem Controller ja sagen, dass ich ihm gleich den Saft klaue und er doch bitteschön sich die aktuellen Werte (im EEPROM sic) merken möge, damit wir beim nächsten Einschalten genau da weitermachen können, wo wir aufgehört haben.

## 1.5 Weiterer Aufbau der Anleitung

In den Kapiteln 2 bis 5 werden die einzelnen Menüs vorgestellt, im Wesentlichen in der Reihenfolge des Auftretens oder der Priorität.

Im Kapitel 6 gibt es noch ein paar Hintergrundinformationen.

## 2 Init

Hier startet unser Uhrtaktgeber seinen Betrieb, dabei sind wir eher in einer Statistenrolle, allerdings nicht ganz.

### 2.1 „Begrüßung“

```
Taktgeber 1.0B00  
Enter hilft
```

Dieser „Menüpunkt liegt eigentlich noch vor dem Init und dient im Wesentlichen Entwicklungszwecken. Also ignorieren und das tun was dort steht: „Enter“ drücken, und der Taktgeber startet.

### 2.2 Init (Systemstatus)

```
SysStatus  Init  
-E----GD   Norml
```

Jetzt hat der Taktgeber seine Startroutine durchlaufen, die Werte aus dem EEPROM geholt und versucht das Beste daraus zu machen. Die einzelnen Statusbits haben, wenn sie angezeigt werden folgende Bedeutung:

- E: EEPROM nicht initialisiert. Das sollte **nur** nach einem Softwareupdate passieren! Sonst liegt ein echter Fehler vor, kaputte Stromversorgung oder so.
- G: „Got dirty bit“. Meint, der letzte Benutzer hat den Taktgeber nicht ordentlich mit „Shutd“ heruntergefahren, also die (da) aktuellen Werte nicht gespeichert. Uhren und Werte nachstellen und letzten Benutzer ausschimpfen (schau mal in den Spiegel;-)
- D: „Dirty bit“. Wird in Init gesetzt und bedeutet, die aktuellen Werte sind gegenüber EEPROM verändert, muß vor dem Ausschalten mit „Shutd“ gesichert werden. Erscheint nur D, ist das gut und es gibt als Vorbelegung im Zielmenü „Norml“. So sollte es sein.

### 3 Norml (Normalbetrieb)

#### 3.1 Einstieg

03:10 10,1 Norml  
11:27 NmUhr

So sieht das Einsprungmenü für den Normalbetrieb aus. Der Normale Uhrbetrieb ist vorbelegt Mit „Up“/„Down“ kann ich folgende Zielmenüpunkte einstellen und mit „Enter“ dann auswählen:

- NmUhr: Normaler Uhrbetrieb (Uhr läuft, Uhr steht“)
- NmSvr: Schneller Vorlauf bis....
- NmVkz: Zeitverkürzung einstellen
- NmSZt: Startzeit einstellen
- Progr: Springe in das Programmiermenü (siehe dort)
- Shutd: Springe ins Shutdownmenü (siehe dort)

Die einzelnen Menüpunkte werden im Folgenden beschrieben.

#### 3.2 NmUhr (Normaler Uhrbetrieb)

03:10 10,1 NmUhr  
11:27 Start

Die Uhr ist im Normalbetrieb und wartet auf „Enter“ um mit der eingestellten Verkürzung loszulaufen (Start), oder „Up/Down“ um „Zurck“ auszuwählen.

Nach Enter mit „Start“ erscheint (Uhr läuft!!):

03:10 10,1 NmUhr  
11:27 Stop

Tja, nun wartet sie auf den Stopbefehl. Mit „Enter“ wird er ausgeführt (Uhr steht).

Und das kann ich nun beliebig oft hin- und herschalten. Nur im ersten Punkt kann ich mit „Up/Down“ auch statt „Start“ „Zurck“ auswählen und komme dann wieder zum Einstieg.

#### 3.3 NmSvr (Schneller Vorlauf bis ...)

03:10 NmSvr  
11:27 SvSrt

Hier im schnellen Vorlauf bis... kann man mit „Up/Down“ auswählen und mit „Enter“ erreichen:

- SvSrt: Schneller Vorlauf bis „Startzeit“
- SvNul: Schneller Vorlauf bis „00:00 Uhr“
- Zurck: Eine Menüebenen zurück (also Einstieg)

In „SvSrt“ und „SvNul“ funktioniert die Bedienung exakt genauso, wie bei „NmUhr“ (siehe dort), nur mit zwei kleinen Abweichungen:

- Statt mit der Zeitverkürzung werden jetzt die Minutenpulse alle zwei Sekunden ausgegeben (Schneller Vorlauf sic)
- Die Uhren halten automatisch bei Erreichen der Zielzeit (Startzeit oder 00:00 Uhr) an.

Somit kann ich komfortabel (automatisch) die Uhren vorlaufen lassen bis ... und trotzdem durch Anhalten mit der Hand auch jede andere Zeit erreichen.

### 3.4 NmVkz (Zeitverkürzung)

```

10,1 NmVkz
Up/Down  Zurck

```

Jetzt haben wir eine etwas andere Logik, das zeigt uns auch der Menüaufbau:

- Mit „Up/Down“ stelle ich den Wert der Zeitverkürzung ein, nicht mehr einen Menüpunkt!! Kurzes Drücken zählt einen Wert rauf oder runter, langes Drücken startet die automatische Wiederholung, sprich er zählt flott rauf oder runter. Bei Erreichen der Maximal- oder Minimalwerte wird automatisch Umgebrochen.
- Mit „Enter“ gehe ich zurück und **speicher** den eingestellten Wert.
- Mit „Zurück“ gehe ich zurück und **verwerfe** den eingestellten Wert.

### 3.5 NmSZt (Startzeit)

```

03:10 NmSZt
Minte

```

Wie gewohnt mit „Up/Down“ den gewünschten Menüpunkt einstellen und mit „Enter“ hineinspringen:

- Minte: Startzeit Minute einstellen
- Stnde: Startzeit Stunde einstellen
- Zurck: Zurück ohne etwas zu tun.

## 3.5.1 **Minte**

03:10          NmSZt  
Up/Down      Zurck

Bedienung genau wie bei „NmVtz“ (Zeitverkürzung), nur werden jetzt die Minutenwerte der Startzeit eingestellt.

## 3.5.2 **Stnde**

03:10          NmSZt  
Up/Down      Zurck

Wie „Minte“, nur für die Stundenwerte.

## 3.6 **Kleine Amerkung**

Das war es eigentlich schon, jetzt brauchen wir „nur“ noch ein paar Zusatzfunktionen, die in den Kapiteln 4 und 5 beschrieben werden.

## 4 Progr (Programmierung)

Jetzt sind wir im Programmiermenü. Sehr viel ist hier noch nicht los, was sich aber in späteren Versionen durchaus ändern kann.

Trotzdem brauchen wir diese Funktionen.

### 4.1 Einstieg

```
Progr  
PrPuB
```

Auch hier wieder, wie schon oft geübt; Menü mit „Up/Down“ einstellen und mit „Enter“ reinspringen. Folgende Möglichkeiten stehen uns zur Verfügung:

- PrPuB: Damit kann die Pulsbreite in engen Grenzen nachjustiert werden.
- PrSe0: Setze die aktuelle Uhrzeit auf 00:00 Uhr. Z.B.: Beim Aufbau für ein Treffen
- Norml: Damit komme ich in den Normalbetrieb zurück
- Shutd: Damit springe ich das Menü Shutdown

### 4.2 PrPuB

```
1,0 Sek PrPuB  
Up/Down Zurck
```

Einstellen der Pulsbreite in engen Grenzen. Das sollte eigentlich „nie“ nötig sein, da normgerecht die Pulse exakt 1,0 Sek lang sind.

Bedienung: Wie schon bei „NmVlz“ beschreiben.

### 4.3 PrSe0

```
PrSe0  
07:54 Zurck
```

Mit „Up/Down“ „PrSe0“ oder „Zurck“ einstellen. „Enter“ führt bei „PrSe0“ das Setzen der aktuellen Uhrzeit auf 00:00 Uhr aus. „Zurück“ ist der Notausgang und springt zurück ohne etwas zu tun.

## 5 Shutd (Shutdown)

Dies ist ein kleines, unscheinbares und leicht zu vergessendes Menü, das aber durchaus seine Daseinsberechtigung hat. Hier erkläre ich nämlich dem Mikrocontroller, dass wir ihm den Saft abdrehen wollen, und er doch nun mal alle wichtigen Werte wieder ins EEPROM schreiben soll und es als „sauber“ kennzeichnet.

Damit hier nicht gleich einversehentliches Aufrufen des Menüs zum Abschalten führt, ist eine kleine Spaßbremse eingebaut, eine Sicherheitsabfrage.

### 5.1 Einstieg

```
03:10 10,1 Shutd  
11:27      Zurck
```

Drücken wir jetzt „Enter“ sind wir wieder zurück im Normalbetrieb. Erst nach Einstellen von „Shutd“ mit „Up/Down“ und anschließendem „Enter“ wird die Funktion wirklich ausgeführt, und hält den Uhrtaktgeber an. Ein Neustart ist nur noch über Saft aus, Saft an möglich.

### 5.2 Ende

```
Taktgeber 1.0B00  
EndOfSoap! Shutd
```

Jetzt steht der Taktgeber und der Strom kann ausgeschaltet werden, ohne daß er etwas vergisst. Eigentlich wäre diese Meldung gar nicht nötig, aber etwas Komfort darf schon sein.

## 6 Hintergründiges

Jetzt sind wir durch die eigentliche Bedienung durch. Spielt einfach mal in einer ruhigen Minute mit dem Uhrtaktgeber, bis Ihr die Menüs beherrscht. Kaputt machen kann man (eigentlich) nichts, ausser man schließt ihn falsch an.

Aber jetzt noch ein paar kleine Hintergrundinformationen für Neugierige. Nicht wirklich wichtig, aber eben doch interessant.

### 6.1 Warum „Shutdown“?

Oder: warum kann der sich das nicht einfach immer selber merken?

Nun, so ein EEPROM, das ist ein nicht flüchtiger Speicher, der mehrfach wiederbeschrieben werden kann, hat nur eine begrenzte Lebensdauer. Nach 100000 Schreibzyklen ist es kaputt. Das hört sich zwar viel an, ist es aber nicht. Um nämlich wirklich alles sauber zu speichern müßte ich bestimmte Informationen bei **jedem** Minutenpuls speichern.

Naja, ein Fahrplandurchlauf (24 Stunden) sind leider 1440 Minuten oder eben soviele Schreibzyklen, damit wäre das EEPROM nach nur 70 Fahrplänen hin. Sch...ade.

### 6.2 Was bringen spätere Versionen?

Hmm, Die Hardware für das Loconet ist nicht aus Spaß auf der Platine. Ich träume immer noch vom Betrieb der „Fastclock“ über das Loconet. Mal sehen, wohin sich die Diskussion bewegt.

Sollten irgendwelche wirklich wichtigen Funktionen fehlen, können wir über eine Einbindung reden.

Wer will, kann sich natürlich auch selbst Ergänzungen schreiben, die Software ist Open Source.

### 6.3 Wo finde ich weiter Informationen?

<http://fremocom.sourceforge.net/fremoclock/index.php>

Hier findet Ihr mehr, z.B. den Quellcode etc.

## 7      **Änderungsindex**

Nr.	Datum	Bemerkung
00	17.05.2007	Erstellung
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		
11		

**Tabelle 1    Änderungsindex**