

Bedienungsanleitung Programmgenerator PG 8300(Batch)



DOC-083-01

Programmgenerator PG 8300

1. Einleitung

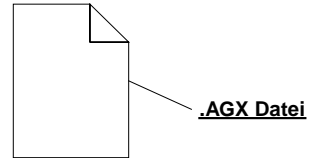
Der Programmgenerator wurde entwickelt, um schnell und einfach ohne jegliche Programmierkenntnisse eigene Anwendungen auf dem PC zu entwickeln. Auf einer Windowsoberfläche lässt sich so in wenigen Minuten eine Anwendung erzeugen, die den meisten Anforderungen genügt. Anschließend kann diese auf das Terminal geladen werden und kann dort sofort auf ihre Tauglichkeit geprüft werden.

Auf den folgenden Seiten finden Sie eine kurze bildliche Darstellung zum grundsätzlichen Prinzips zur Erstellung eigener Anwendungen mit Hilfe des Programmgenerators.

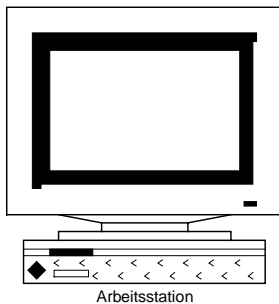
Schritt 1



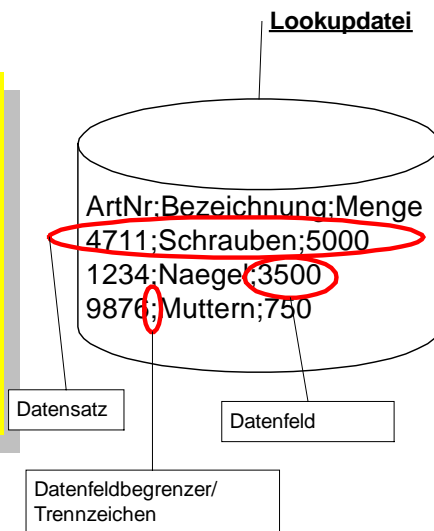
Mit Hilfe des Programmgenerators erstellen Sie Ihre gewünschte Anwendung



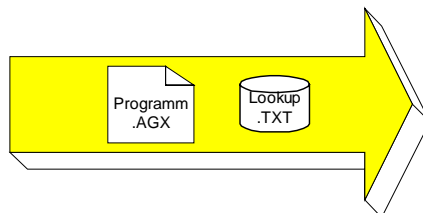
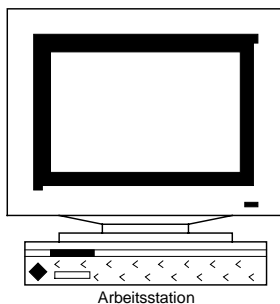
Schritt 2



Am PC erstellen Sie eine Textdatei, die nachher auf dem Terminal in eine indexierte Datenbank umgewandelt wird. Diese Datei wird im folgenden Text als **Lookupdatei** bezeichnet.



Schritt 3

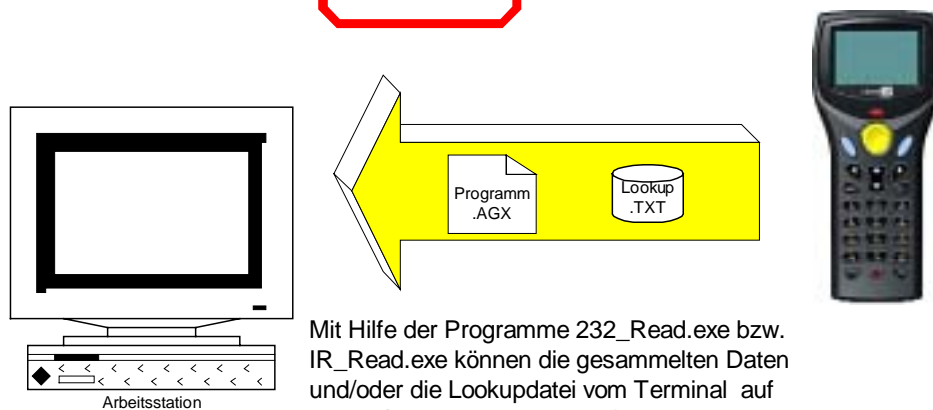


mit Hilfe der Programme AGX_Load.exe bzw. DLookup.exe werden das Programm bzw. die Lookupdatei auf das Terminal geladen (Dies kann auch direkt vom Programmgenerator aus geschehen). Hierzu muss das Terminal entweder direkt über ein Kabel oder das Cradle mit dem PC verbunden sein.

Schritt 4



Schritt 5



Mit Hilfe der Programme 232_Read.exe bzw. IR_Read.exe können die gesammelten Daten und/oder die Lookupdatei vom Terminal auf den PC übertragen werden (Dies kann auch direkt vom Programmgenerator aus geschehen). Hierzu muss das Terminal entweder direkt über ein Kabel oder über das Cradle mit dem PC verbunden sein.

2. Starten des Programmgenerators

Ein Click mit der rechten Maustaste auf das Symbol oder ein Click mit der linken Maustaste auf die POWER-Taste öffnet das Kontextmenü.

Neu	Ctrl+N	- eine neue Anwendung erstellen
Öffnen...	Ctrl+O	- eine bestehende Anwendung (*.AGX) öffnen
Speichern	Ctrl+S	- aktive Anwendung sichern
Speichern unter ...		- aktive Anwendung unter anderem Namen speichern
Bearbeiten		- die aktive Anwendung bearbeiten
COM Port Einstellungen...		- Schnittstelle für die Verbindung zum Terminal festlegen
Programm laden	▶	- die aktive Anwendung auf das Terminal übertragen
Lookupdatei laden	▶	- Lookupdatei auf das Terminal übertragen
Daten empfangen	▶	- Daten vom Terminal an den PC senden
Über...		- Info über den Revisionsstand
Ende	Alt+F4	- Verlassen des Programm-Generators

3. Masken

Bildschirmmasken können aus bis zu acht Eingabefeldern bestehen. Die Eigenschaften jedes einzelnen Feldes können individuell gestaltet werden. Insgesamt können zehn völlig unterschiedliche Masken für Ihre Anwendung definiert werden.

Application Generator

Maske | Benutzermenü | Lookup | Barcodeeinstellungen | Einstellungen | Statmaske

Name: Esc: Lookupdatei: Zeitzusatz:

Schaltart: Nächste Aktion: Datensatz: Grundeinstellungen:

Zelle	Datentyp	Displayanzeige	Datenquelle	Min Länge	Max Länge	Lookup	Optionen
#1	<input type="text" value="deco"/>	<input type="text" value=""/>	<input type="text" value="Beide"/>	<input type="text" value="0"/>	<input type="text" value="50"/>	<input type="text" value="deco"/>	<input type="text" value="mehr..."/>
#2	<input type="text" value="deco"/>	<input type="text" value=""/>	<input type="text" value="Beide"/>	<input type="text" value="0"/>	<input type="text" value="50"/>	<input type="text" value="deco"/>	<input type="text" value="mehr..."/>
#3	<input type="text" value="deco"/>	<input type="text" value=""/>	<input type="text" value="Beide"/>	<input type="text" value="0"/>	<input type="text" value="50"/>	<input type="text" value="deco"/>	<input type="text" value="mehr..."/>
#4	<input type="text" value="deco"/>	<input type="text" value=""/>	<input type="text" value="Beide"/>	<input type="text" value="0"/>	<input type="text" value="50"/>	<input type="text" value="deco"/>	<input type="text" value="mehr..."/>
#5	<input type="text" value="deco"/>	<input type="text" value=""/>	<input type="text" value="Beide"/>	<input type="text" value="0"/>	<input type="text" value="50"/>	<input type="text" value="deco"/>	<input type="text" value="mehr..."/>
#6	<input type="text" value="deco"/>	<input type="text" value=""/>	<input type="text" value="Beide"/>	<input type="text" value="0"/>	<input type="text" value="50"/>	<input type="text" value="deco"/>	<input type="text" value="mehr..."/>
#7	<input type="text" value="deco"/>	<input type="text" value=""/>	<input type="text" value="Beide"/>	<input type="text" value="0"/>	<input type="text" value="50"/>	<input type="text" value="deco"/>	<input type="text" value="mehr..."/>
#8	<input type="text" value="deco"/>	<input type="text" value=""/>	<input type="text" value="Beide"/>	<input type="text" value="0"/>	<input type="text" value="50"/>	<input type="text" value="deco"/>	<input type="text" value="mehr..."/>

OK Abbrechen

3.1 Eigenschaften einer Maske

3.1.1. Schriftart

Bei großer Schriftart werden 4x15, bei kleiner Schriftart 8x20 Zeichen auf dem Display dargestellt.

3.1.2. Esc

Festlegen, was dargestellt werden soll, wenn die ESC-Taste gedrückt wird.

3.1.3. Nächste Aktion

Festlegen, was dargestellt werden soll, wenn der Benutzer die Dateneingabe für diese Maske beendet hat.

3.1.4. Lookupdatei

Festlegen, welche Lookupdatei für diese Maske benutzt wird. Es können bis zu drei Lookupdateien definiert werden.

3.1.5. Datensatz

Festlegen, was zu tun ist, wenn der Benutzer die Dateneingabe für eine Maske beendet hat.

- **Speichern:** Speichert den eingelesenen Datensatz.
- **update lookup:** Nur die ausgewählte Lookupdatei wird aktualisiert.
- **save & update:** Der eingelesene Datensatz wird gespeichert und die ausgewählte Lookupdatei wird aktualisiert.
- **pass down:** Wird benutzt, um Daten in der nächsten Maske erneut darzustellen, ohne diese zu speichern. Dafür muss in der folgenden Maske der Datentyp pass down ausgewählt werden.
- **output record:** Datensatz wird nicht gespeichert, sondern direkt an den COM-Port geschickt.
- **output screen** Es wird die komplette Maske dargestellt, inklusive Überschriften. Datensatz wird nicht gespeichert.

3.2 Eigenschaften der einzelnen Felder einer Maske

Die Eigenschaften für jedes Feld einer Maske sind folgende:

3.2.1. Datentyp

- **<leer>:** Feld wird nicht benutzt
- **Text:** alle Zeichen (z.B. \$1a2b3c*+/-/...)
- **Ganzzahl:** nur ganze Zahlen (z.B. 12345)
- **Realzahl:** Kommazahlen (z.B. 3.14159)
- **Buchstaben:** nur Alphabet (z. B. aBcDE)
- **Boolean:** Es kann nur '0' oder '1', 'N' oder 'Y' eingegeben werden.
- **Lookup:** Daten werden nicht vom Benutzer eingegeben, sondern kommen von einem vorher bestimmten Feld aus der Datenbank, sofern der Datensatz im vorherigen Feld mit dem Schlüsselfeld der Datenbank übereinstimmt.
- **fixe Daten:** keine Eingabe, dieses Feld wird dem Datensatz angehängt.
- **Überschrift:** keine Eingabe, nur Darstellung auf dem Display.
- **Zähler:** Zähler wird auf dem Display dargestellt und bei jedem neu eingelesenen Datensatz um 1 erhöht.
- **pass down:** Wenn in der vorherigen Maske unter Datensatz "pass down" ausgewählt wurde werden nun alle Daten aus der vorherigen Maske dargestellt und mit jedem Datensatz abgespeichert.
- **Erweiterung:** Erweiterungsfeld eines vorherigen Datenfeldes, falls die Zeile dort nicht ausgereicht hat (z. B. 20-stelliger Barcode).

3.2.2. Displayanzeige

Festlegen, was auf dem Display angezeigt wird.

3.2.3. Datenquelle

Festlegen, von welcher Datenquelle das Feld gefüllt werden darf (Scanner, Tastatur oder Beide).

3.2.4. Min. Länge

Die minimale Anzahl der einzugebenden Zeichen.

3.2.5. Max. Länge

Die maximale Anzahl der einzugebenden Zeichen. Die maximale Länge kann bis zu 50 Stellen betragen. (**Achtung:** maximale Stellenanzahl einer Zeile im Display ist 20. Sollen mehr Zeichen dargestellt werden, muss in der nächsten Zeile Erweiterung ausgewählt werden).

3.2.6. Lookupdatei

Festlegen der Lookupdatei, auf die sich das einzugebende Datenfeld bezieht.

3.2.7. Optionen (mehr Eigenschaften)

Felddaten

- *Feste Feldlänge:* Das Feld soll auf eine bestimmte Länge gebracht werden. Zusätzlich kann die Ausrichtung eingestellt werden und mit welchen Zeichen die entstehenden Leerstellen gefüllt werden sollen.
- *Voreinstellung:* Der hier eingegebene Wert wird als "Default" auf dem Display dargestellt.
- *Präfix:* Durch Anklicken des Feldes öffnet sich ein Fenster, in dem bis zu zehn verschiedene Zeichen ausgewählt werden können, die vor die eingegebenen Daten gesetzt werden.
- *Suffix:* Durch Anklicken des Feldes öffnet sich ein Fenster, in dem bis zu zehn verschiedene Zeichen ausgewählt werden können, die hinter die eingegebenen Daten gesetzt werden.
- *Füllzeichen:* Die Zeile auf dem Display wird mit dem eingestellten Zeichen gefüllt (Das Zeichen wird nicht gespeichert).

Barcodedaten

- *Teilstring aus Barcode:* nur einen Teil des Barcodes einlesen
- *Startposition:* Ab welcher Stelle sollen die eingelesenen Daten übernommen werden
- *Anzahl Zeichen:* Anzahl der Zeichen, die nach der Startposition übernommen werden sollen
- *Präfix prüfen:* Hier kann geprüft werden, ob der eingelesene Wert die hier eingestellten führenden Zeichen beinhaltet.
- *AUTO Übernahme:*
 - Nein: Benutzer muss die Enter-Taste betätigen, um Datensatz zu speichern.
 - Scan+ENTER: Datensatz wird direkt nach dem Scannen gespeichert und verschwindet vom Display.

4. Benutzermenü

Ein Hauptmenü, um die auszuführenden Masken auszuwählen.

The screenshot shows the 'Application Generator' window with the 'Benutzermenü' tab selected. The window contains the following elements:

- Navigation tabs:** Maske, Benutzermenü, Lookup, Barcodeeinstellungen, Einstellungen, Startmaske.
- General settings:**
 - Name:
 - Schriftart:
 - Esc:
 - ☐ Menü-Überschrift:
- Menu items table:**

Zeile	Anzeigetext	Sprung zu
#1	<input type="text"/>	<input type="text" value="Startmaske"/>
#2	<input type="text"/>	<input type="text" value="Startmaske"/>
#3	<input type="text"/>	<input type="text" value="Startmaske"/>
#4	<input type="text"/>	<input type="text" value="Startmaske"/>
#5	<input type="text"/>	<input type="text" value="Startmaske"/>
#6	<input type="text"/>	<input type="text" value="Startmaske"/>
#7	<input type="text"/>	<input type="text" value="Startmaske"/>
#8	<input type="text"/>	<input type="text" value="Startmaske"/>
#9	<input type="text"/>	<input type="text" value="Startmaske"/>
#10	<input type="text"/>	<input type="text" value="Startmaske"/>
- Daten section:**
 - ☐ Überschrift speichern (als einzelnen Datensatz)
 - ☐ Menüpunkt speichern (als einzelnen Datensatz)
 - ☐ Menüpunkt speichern (in jedem Datensatz)
- Footer:**
 - ☐ gesammelte Daten für jede Maske in eine separate Datei speichern
 -
 -
 -

- *Schriftart:* Bei großer Schriftart werden 4x15, bei kleiner 8x20 Zeichen auf dem Display dargestellt
- *Menü-Überschrift:* Überschrift, die im Display über dem Benutzermenü dargestellt wird.
- *Anzeigetext:* Anzeige der einzelnen Menüpunkte
- *Sprung zu:* Festlegen, zu welcher Maske bei Anwahl dieses Menüpunktes gesprungen werden soll
- *Daten:*
 - Überschrift speichern: Speichert die Menü-Überschrift als einzelnen Datensatz.
 - Menüpunkt speichern(einzeln): Speichert den Menüpunkt als einzelnen Datensatz. Nach Empfang der Datei am PC kann man erkennen welchen Datensatz der Benutzer unter welchem Menüpunkt gespeichert hat.
 - Menüpunkt speichern(jeden): Speichert den Menüpunkt vor jedem Datensatz.

5. Lookup

Eine Lookupdatei ist eine Datenbank, auf die bei der Arbeit in den Masken zugegriffen werden kann. Datenfelder aus einer Lookupdatei können *nur angezeigt*, aber auch *verändert* werden. Es können bis zu drei Lookupdateien verwaltet werden. Eine Lookupdatei enthält folgende Eigenschaften:

The screenshot shows the 'Programmengenerator PG8000' window with the 'Lookup' tab selected. The 'Name' dropdown is set to '1. Lookupdatei'. The 'Feldlänge' is 0 and 'Anzahl Felder' is 0. Under 'Feldereigenschaften', 'feste Länge' is selected. A table lists 8 fields with their positions and lengths. At the bottom, there are buttons for 'OK', 'Abbrechen', and 'Grundeinstellungen'.

Feld	ab Position	Länge	Schlüsselfeld
#1	1	0	<input type="radio"/>
#2	1	0	<input type="radio"/>
#3	1	0	<input type="radio"/>
#4	1	0	<input type="radio"/>
#5	1	0	<input type="radio"/>
#6	1	0	<input type="radio"/>
#7	1	0	<input type="radio"/>
#8	1	0	<input type="radio"/>

5.1. Anzahl Felder

Festlegen, aus wie vielen Feldern ein Datensatz dieser Lookupdatei besteht.

5.2. Feldeigenschaften

feste Länge: Festlegen, ob ein Feld eines Datensatzes eine jeweils fest definierte Länge hat.

Trennzeichen: Ein Feld hat in den Datensätzen unterschiedliche Längen, grenzt sich aber durch ein spezielles Zeichen von seinen Nachbarfeldern im Datensatz ab.
(ASCII-Wert)

5.3. ab Postition

Startposition für jedes Feld der Lookupdatei (nur wenn "feste Länge" ausgewählt wurde aktiv).

5.4. Länge/ Max. Länge

Definiert die Länge des Datenfeldes, falls "feste Länge" ausgewählt wurde.

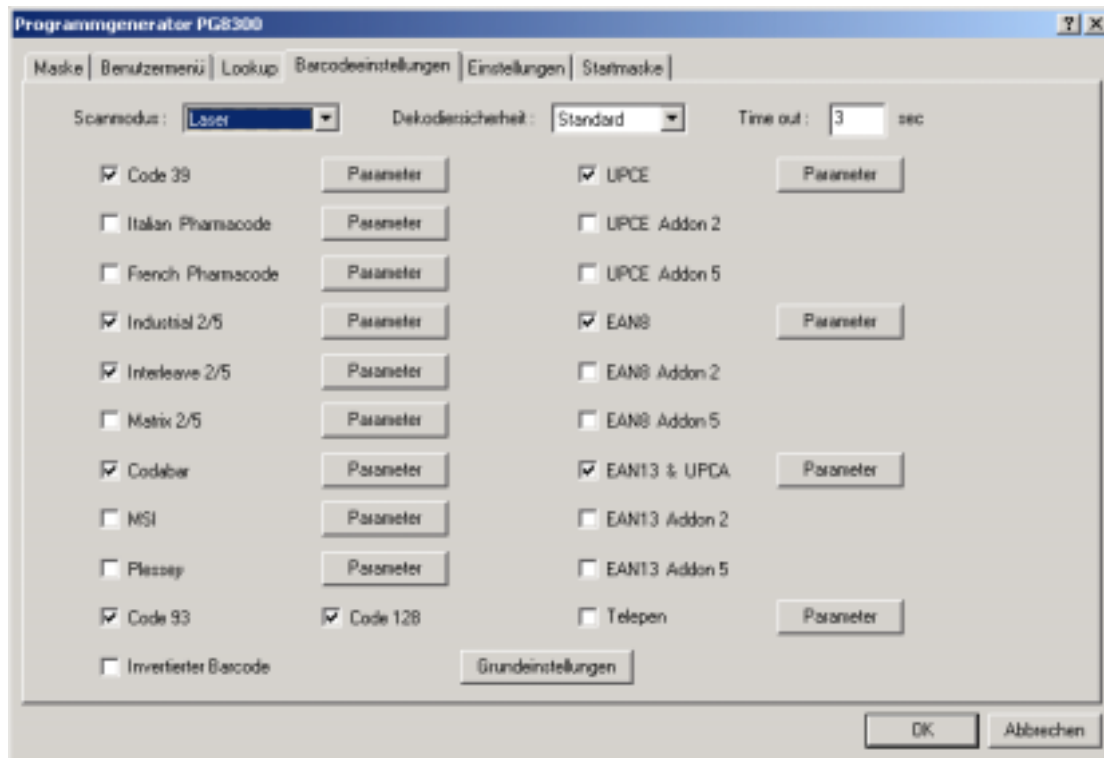
Definiert die maximale Länge für das Datenfeld, falls "Trennzeichen" ausgewählt und definiert wurde.

5.5. Schlüsselfeld

Das Schlüsselfeld ist das Feld, das als Referenz zum Auffinden eines Datensatzes benutzt wird. Wenn sich ein Eingabefeld auf ein Schlüsselfeld bezieht, werden die anderen Daten auf dem Display angezeigt (falls Datentyp "Lookup" für ein Feld definiert wurde), sobald die eingegebenen Daten mit den Schlüsselfelddaten übereinstimmen.

6.Barcodeeinstellungen

Hier können benutzerdefinierte Einstellungen der einzelnen Barcodefamilien eingestellt werden.



6.1. Aktivieren/ Deaktivieren der einzelnen Codefamilien

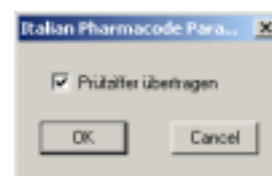
6.1.1. Code39

- **Start/Stop Zeichen übertragen** : Einstellen, ob Start/Stop-Zeichen mit dem Barcode übertragen werden soll.
- **Prüfziffer berechnen** : Auswahl, ob die Prüfziffer beim Dekodieren berechnet werden soll. Sollte die Prüfziffer nicht korrekt sein, wird der Barcode nicht gelesen.
- **Prüfziffer übertragen** : Auswahl, ob die Prüfziffer mit dem Barcode übertragen werden soll.
- **Standard / Full ASCII Code 39** : Benutzer kann auswählen, ob der Standard Code 39 oder der alphanumerische Code 39 benutzt werden soll.



6.1.2. Italian / French Pharmacode

Im italienischen bzw. französischen Pharmacode ist immer eine Prüfziffer enthalten. Deshalb wird die Prüfziffer immer beim Dekodieren berechnet. Der Benutzer kann hier festlegen, ob die Prüfziffer im übertragenen Barcode enthalten sein soll.



6.1.3. Industrial / Interleave / Matrix 2/5

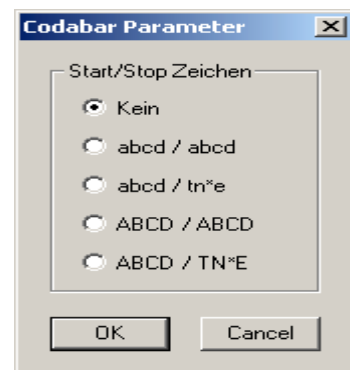
- **Start / Stop Auswahl** : Dieser Parameter stellt die Lesbarkeit aller Varianten der "2 aus 5"-Familie sicher. Ein Beispiel: Auf Flugtickets wird der Industrial 2/5 benutzt, allerdings mit Interleave 2/5 Start/Stop. Um diesen Barcode lesen zu können muss der Start/Stop-Parameter des Industrial 2/5 auf Interleave 2/5 gesetzt sein.



- **Prüfziffer berechnen** : Auswahl, ob die Prüfziffer beim Dekodieren berechnet werden soll. Sollte die Prüfziffer nicht korrekt sein, wird der Barcode nicht gelesen.
- **Prüfziffer übertragen** : Auswahl, ob die Prüfziffer mit dem Barcode übertragen werden soll.
- **Längenbestimmung** : Aufgrund der labilen Struktur des 2/5 Code kann es leicht vorkommen, dass nur ein Teil des Barcodes gelesen wird. Um die Sicherheit zu erhöhen, dass auch der richtige Code gelesen wird, kann über die Längenbestimmung festgelegt werden, dass nur bestimmte Barcodelängen gelesen werden sollen. Wird eine feste Längeneinstellung ausgewählt, können nur Barcodes mit dieser Stellenanzahl gelesen werden. Wird Max/Min Länge ausgewählt, werden nur Barcodes akzeptiert, deren Länge zwischen den beiden Werten liegen.

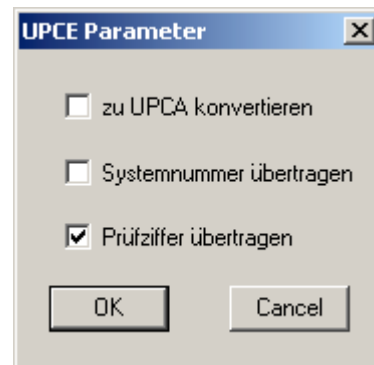
6.1.4. Codabar

- **Start/Stop Zeichen übertragen** : Einstellen, ob Start/Stop-Zeichen mit dem Barcode übertragen werden soll.
- **Start / Stop Selektion** : Es können vier verschiedene Start/Stop-Zeichen-Paare definiert werden:
 abcd / abcd
 abcd / tn*e
 ABCD / ABCD
 ABCD / TN*E



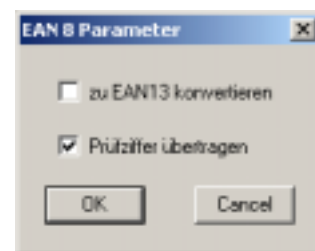
6.1.5. UPCE

- **Zu UPCA konvertieren** : Wenn dieser Parameter ausgewählt wird, wird der gelesene UPCE zu UPCA erweitert. In der weiteren Verarbeitung werden die Parameter des UPCA benutzt.
- **Systemnummer übertragen** : Ist dieser Parameter ausgewählt, wird die UPCE-Systemnummer mit dem eingelesenen Datensatz übertragen.
- **Prüfziffer übertragen** : Auswahl, ob die Prüfziffer mit dem Barcode übertragen werden soll.



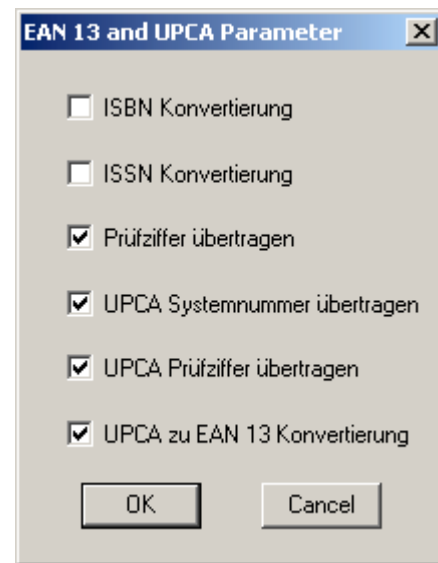
6.1.6. EAN8

- **Zu EAN13 konvertieren** : Wenn dieser Parameter ausgewählt wird, wird der gelesene EAN8 zu EAN13 erweitert. In der weiteren Verarbeitung werden die Parameter des EAN13 benutzt.
- **Prüfziffer übertragen** : Auswahl, ob die Prüfziffer mit dem Barcode übertragen werden soll.



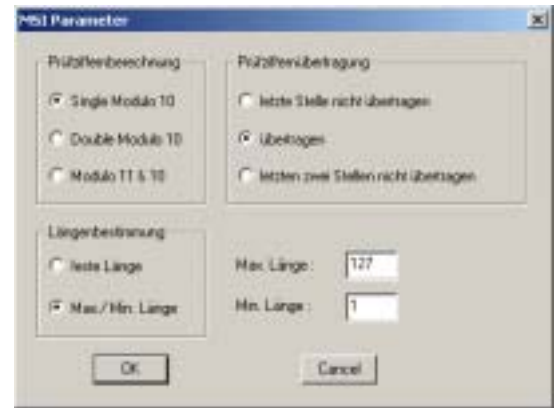
6.1.7. EAN13 & UPCA

- **ISBN / ISSN Konvertierung** : Sollte dieser Parameter ausgewählt sein, konvertiert der Scanner den eingelesenen Code in ISBN oder ISSN-Code, sofern die Formate korrekt sind (EAN13-Codes beginnen mit 978 oder 979 für ISBN, bzw. 977 für ISSN)
- **Prüfziffer übertragen** : Auswahl, ob die Prüfziffer mit dem Barcode übertragen werden soll.
- **UPCA Systemnummer übertragen** : Ist dieser Parameter ausgewählt, wird die UPCA-Systemnummer mit dem eingelesenen Barcode übertragen.
- **Prüfziffer übertragen** : Auswahl, ob die UPCA-Prüfziffer mit dem Barcode übertragen werden soll.
- **Zu EAN13 konvertieren** : Wenn dieser Parameter ausgewählt wird, wird der gelesene UPCA-Code zu EAN13 konvertiert. In der weiteren Verarbeitung werden die Parameter des EAN13 benutzt



6.1.9. MSI

- **Prüfziffernberechnung** : Drei verschiedene Varianten der Prüfziffernberechnung können im MSI-Code enthalten sein: **Single Modulo 10**, **Double Modulo 10**, or **Modulo 11 & 10**. Falls die Prüfziffer nicht korrekt ist, wird der Barcode nicht gelesen.
- **Prüfziffernübertragung** : Die Übertragung der Prüfziffer kann wie folgt eingestellt werden.
 - 1) letzte Stelle nicht übertragen
 - 2) übertragen
 - 3) letzten zwei Stellen nicht übertragen
- **Längenbestimmung** : Aufgrund der labilen Struktur des MSI Code kann es leicht vorkommen, dass nur ein Teil des Barcodes gelesen wird. Um die Sicherheit zu erhöhen, dass auch der richtige Code gelesen wird, kann über die Längenbestimmung festgelegt werden, dass nur bestimmte Barcodelängen gelesen werden sollen. Wird eine feste Längeneinstellung ausgewählt, können nur Barcodes mit dieser Stellenanzahl gelesen werden. Wird Max/Min Länge ausgewählt, werden alle Barcodes, deren Länge zwischen den beiden Werten liegen, akzeptiert.



6.1.10. Plessey

- **UK Plessey Konvertierung**: Falls dieser Parameter ausgewählt ist, ersetzt der Dekoder das Zeichen 'A' durch das Zeichen 'X'.
- **Prüfziffer übertragen** : Auswahl, ob die Prüfziffer (zwei Zeichen) mit dem Barcode übertragen werden soll.



6.2. Scanmodus

Es gibt neun verschiedenen Scanmodi. Der Benutzer kann den gewünschten Scanmodus an seine Anforderungen anpassen. Die enthaltenen Scanmodi haben folgende Bedeutungen:

Auto Off Der Strahl schaltet sich bei Drücken des Triggerknopfes ein und geht nach erfolgreicher Decodierung aus

Continuous Der Strahl schaltet sich beim Programmstart automatisch ein und geht nach erfolgreicher Decodierung nicht aus

Auto Power Off Der Strahl schaltet sich bei Drücken des Triggerknopfes ein und geht nach erfolgreicher Decodierung aus

Alternate Der Strahl schaltet sich bei Drücken des Triggerknopfes ein und geht nach erfolgreicher Decodierung oder nach erneutem Drücken des Triggerknopfes aus

Momentary Der Laserstrahl bleibt solange an, wie der Trigger gedrückt wird

Repeat Der Strahl schaltet sich bei Drücken des Triggerknopfes ein und geht nach erfolgreicher Decodierung nicht aus

Laser Der Laserstrahl bleibt solange an, wie der Trigger gedrückt wird

Test Der Strahl schaltet sich automatisch ein und geht nach erfolgreicher Decodierung nicht aus

Aiming Hier muss der Benutzer zur Dekodierung den Trigger zweimal drücken. Beim ersten Betätigen wird der Code nur anvisiert. Nun muss der Benutzer innerhalb von einer Sekunde den Trigger noch einmal betätigen, um den Barcode zu dekodieren. Geschieht dies nicht innerhalb von einer Sekunde, muss das Ziel noch einmal anvisiert werden. Dieser Modus wird benutzt, wenn zwei Codes sehr eng aneinander gedruckt sind, um sicher zu stellen, dass der Benutzer auch den richtigen anvisierten Code einliest.

6.3. Dekodiersicherheit

Hier kann die Dekodiersicherheit eingestellt werden. Sollte Standard oder einfach ausgewählt sein, benötigt das Terminal nur eine erfolgreiche Dekodierung, um die Lesung gültig zu machen. Sollte zweifach bzw. dreifach ausgewählt sein, müssen zwei bzw. drei erfolgreiche Dekodierungen erfolgen, um die Lesung gültig zu machen. Durch die hohe Dekodiersicherheit benötigt der Scanner mehr Zeit, wird dadurch eventuell etwas langsamer und verbraucht natürlich auch mehr Energie. Der Benutzer muss sich hier zwischen höherer Sicherheit oder besserer Performance entscheiden.

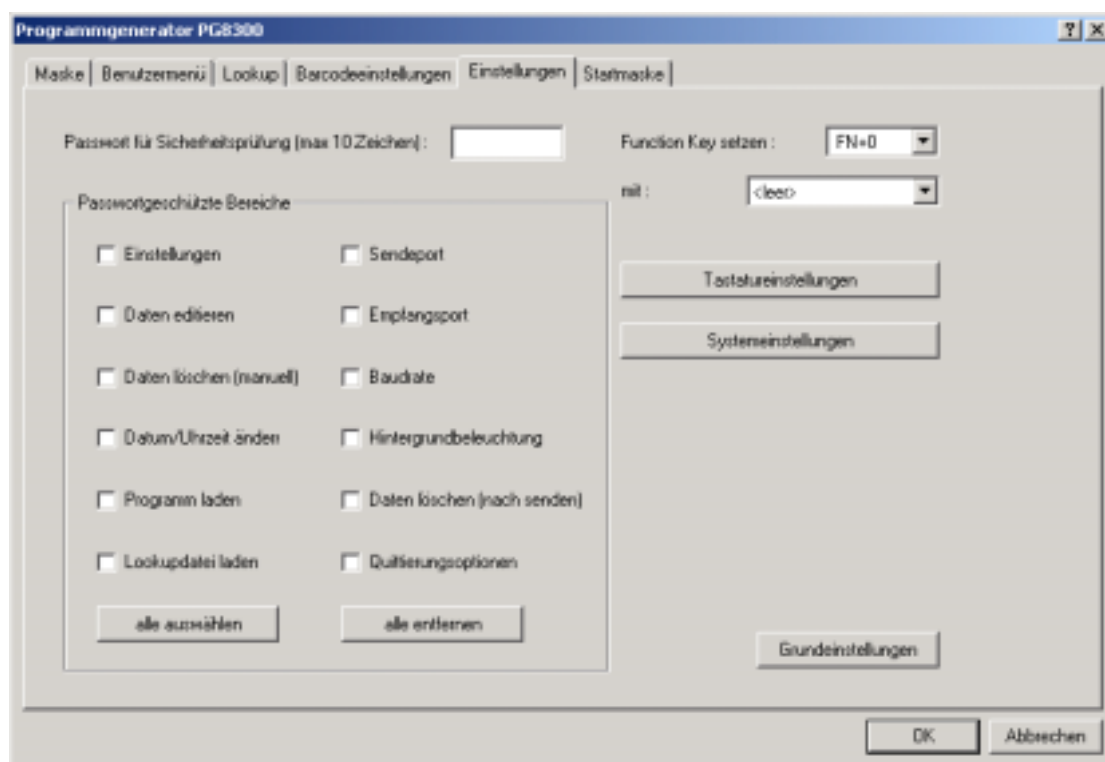
6.4. Time out

Dieser Parameter limitiert die maximale Zeitspanne einer Scanperiode, wenn der Scanmodus AUTO-OFF oder AUTO-POWER-OFF ausgewählt wurde.

6.5. Invertierter Barcode

Der Scanner kann so konfiguriert werden, dass er invertiert gedruckte Barcodes lesen kann. Normalerweise sind Barcodes so gestaltet, dass die Balken dunkler sind, als die Lücken. Invertierte Barcodes sind umgekehrt gestaltet.

7. Einstellungen



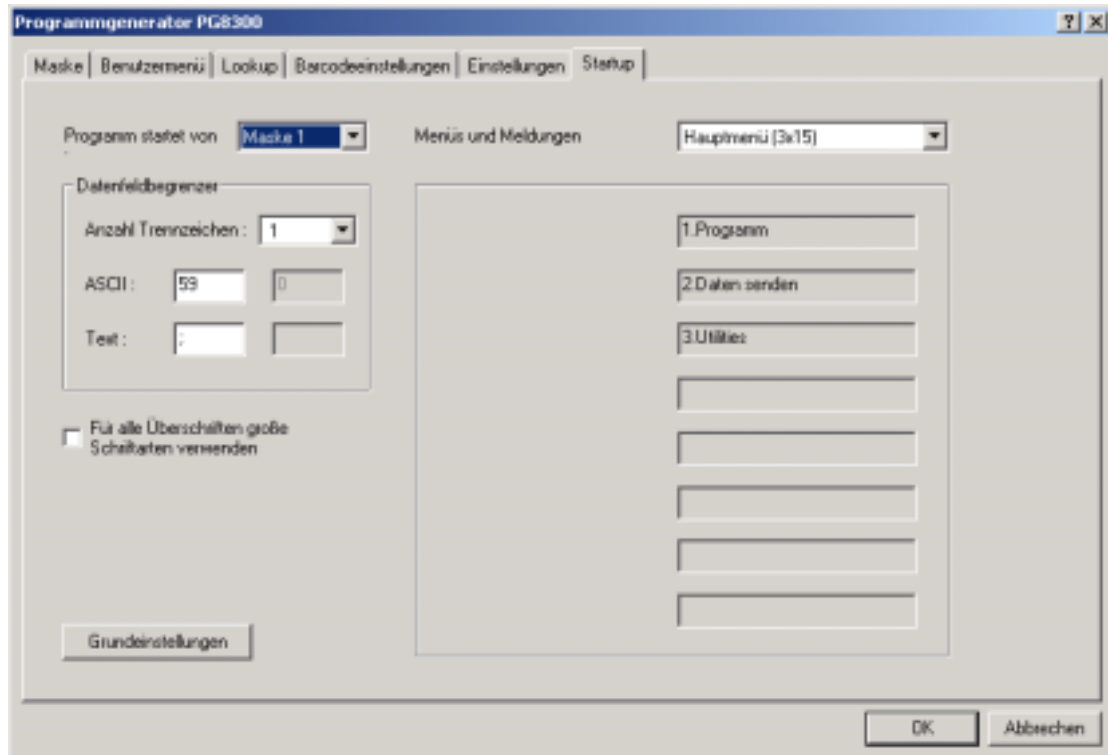
Hier kann ein Passwort hinterlegt werden. Der Entwickler kann entscheiden, für welche Funktion des Terminals eine Passwortabfrage benötigt wird.

Function Key setzen: Hier können die Tasten '0' bis '9' in Verbindung mit der 'FN'-Taste mit einer bestimmten Funktion belegt werden.

Systemeinstellungen: Hier können Systemeinstellungen, wie z.B. Sende- oder Empfangsport eingestellt werden. Diese Einstellungen können auch über "Utilities" auf dem Terminal eingestellt werden.

Tastatureinstellungen: Hier kann der jeweils angeschlossene Tastaturtyp ausgewählt und dessen Einstellungen verändert werden.

8. STARTUP



8.1. Programm startet von:

Hier wird festgelegt, von welcher Maske oder Menü das Programm starten soll.

8.2 Datenfeldbegrenzer

Hier können 0 bis 2 ASCII-Zeichen ausgewählt werden, die die Felder eines Datensatzes trennen sollen.

[zurück](#)

000	NUL	033	!	066	B	099	c	132	ä	165	ñ	198	ä	231	þ
001	Start Of Header	034	"	067	C	100	d	133	å	166	*	199	Å	232	þ
002	Start Of Text	035	#	068	D	101	e	134	ä	167	*	200	ä	233	ú
003	End Of Text	036	\$	069	E	102	f	135	ç	168	¿	201	£	234	ü
004	End Of Transmission	037	%	070	F	103	g	136	é	169	®	202	£	235	ü
005	Enquiry	038	&	071	G	104	h	137	é	170	~	203	¥	236	ý
006	Acknowledge	039		072	H	105	i	138	é	171	%	204	¥	237	ý
007	Bel	040	(073	I	106	j	139	í	172	%	205	—	238	—
008	Backspace	041)	074	J	107	k	140	î	173	í	206	¢	239	~
009	Horizontal Tab	042	*	075	K	108	l	141	í	174	<	207	¤	240	-
010	Line Feed	043	+	076	L	109	m	142	Ä	175	>	208	ä	241	±
011	Vertical Tab	044	,	077	M	110	n	143	Ä	176	∴	209	ø	242	„
012	Form Feed	045	-	078	N	111	o	144	É	177	∴	210	É	243	%
013	Carriage Return	046	.	079	O	112	p	145	œ	178	■	211	É	244	†
014	Shift Out	047	/	080	P	113	q	146	Æ	179		212	É	245	§
015	Shift In	048	0	081	Q	114	r	147	ö	180		213	í	246	+
016	Delete	049	1	082	R	115	s	148	ö	181	Ä	214	í	247	„
017	-- frei --	050	2	083	S	116	t	149	ö	182	Ä	215	í	248	*
018	-- frei --	051	3	084	T	117	u	150	ü	183	Ä	216	í	249	~
019	-- frei --	052	4	085	U	118	v	151	ü	184	®	217	J	250	-
020	-- frei --	053	5	086	V	119	w	152	ý	185	§	218	r	251	*
021	Negative Acknowledge	054	6	087	W	120	x	153	Ö	186		219	■	252	*
022	Synchronous Idle	055	7	088	X	121	y	154	U	187	§	220	■	253	*
023	End Of Transmission Block	056	8	089	Y	122	z	155	ø	188	§	221	í	254	■
024	Cancel	057	9	090	Z	123	{	156	£	189	§	222	í	255	
025	End Of Medium	058	:	091	[124		157	ø	190	¥	223	■		
026	Substitute	059	;	092	\	125	}	158	×	191	γ	224	ó		
027	Escape	060	<	093]	126	~	159	f	192	L	225	ß		
028	File Separator	061	=	094	^	127	o	160	ä	193	⊥	226	ó		
029	Group Separator	062	>	095	_	128	ç	161	í	194	T	227	ó		
030	Record Separator	063	?	096	`	129	ü	162	ä	195	†	228	ä		
031	Unit Separator	064	@	097	a	130	ü	163	ü	196		229	ü		
032		065	^	098	b	131	ä	164	ä	197	+	230	ü		

Das Paket der 8300er Serie enthält folgende Dateien

- PG8300.EXE: Programmgenerator
- Demo-8300.AGX: Ein Demoprogramm
- G8300-xxx.SHX: Die "runtime" für den PG
- K8300-xxx.SHX: Der Kernel für die 8000er Serie
- 232_Read.EXE: zur Kommunikation mit der seriellen Schnittstelle
- AGX_LOAD.EXE: zum Laden von Programmen, die mit dem
Programmgenerator erstellt wurden
- DLookup.EXE: zum Laden von Lookupdateien
- download.EXE: zum Laden von Kernel und runtime via RS232 oder IrDA
- IR_Load.EXE: zum Laden via Cradle bzw. IR-Transceiver
- IR-Read.EXE: zum Empfangen von Daten über Cradle und
IR-Transceiver
- RF-DB-1.TXT: Lookupdatei für Programmpunkt "3.Lookup Demo"