



15.08.2016

Programmierhandbuch

PM String Change

PSi Drucker Familie PP 40x/PP 80x

Acknowledgement

EPSON is a trademark of the Seiko Epson Corporation.

IBM, ProPrinter are trademarks of the International Business Machines Corporation.

Microsoft, Windows, Windows NT, are trademarks of Microsoft Corporation

A Publication of PSi Matrix GmbH
Hommewiese 116c
D – 57258 Freudenberg
Federal Republic of Germany
November 2015
<http://www.psi-matrix.eu>

Great care has been taken to ensure that the information in this handbook is accurate and complete. However, should any errors or omissions be discovered or should any user wish to make suggestions for improving this handbook, please feel encouraged to send us the relevant details.

The contents of this manual are subject to change without notice.

Copyright © 2016 by Psi Matrix GmbH.

All rights strictly reserved. Reproduction or issue to third parties in any form is not permitted without written authorization from the publisher.

Inhalt

1	Beschreibung der PM's mit StringChange Funktion	4
1.1	Personality Modules mit der Funktion StringChange	4
2.1	Installation des PM's.....	1
3.1	Stecker, Taster und LED- Anzeige	1
2	Anwendung und Installation.....	2
2.1	Programmierung.....	2
3	Definition der StringChange-Funktionen	3
3.1	StringChange – Funktionen Mode 0 bis Mode 8	4
3.2	Format einer StringChange – Definition:	5
3.3	Anwendung von Jokern und Übernahmezeichen.....	5
3.4	Welcher Trigger wird wirksam?	6
4	Beschreibungen und Beispiele der einzelnen Modi.....	6
4.1	Mode 0.....	6
4.2	Mode 1.....	7
4.3	Mode 2.....	8
4.4	Mode 3.....	9
4.5	Mode 4.....	10
4.6	Mode 5.....	10
4.7	Mode 7.....	12
4.8	Mode 8.....	13
5	Allgemeine Hinweise.....	15
5.1	Bearbeiten einer STGCHG_D.ini Datei:	15
5.2	Inhalt einer STGCHG_D.ini- Datei	15
5.3	Technische Daten.....	19

Dieser Anhang enthält grundlegende Informationen über die String Change Funktion für alle Modelle Befehle in diesen Druckertypen unterstützt:



1 Beschreibung der PM's mit StringChange Funktion

Ein Personality Modul (PM) mit StringChange- Funktion findet Anwendung, wenn sich Probleme im Druckerdatenstrom ergeben, die durch Anpassungen bzw. Umwandlungen des Datenstroms gelöst werden können. Probleme dieser Art treten auf, wenn vorhandene Drucklösungen modernisiert werden sollen und im Datenstrom Unverträglichkeiten auftreten, die systemseitig nicht abgefangen werden können.

Die Funktion StringChange in den entsprechenden Personality Modules für die Druckerfamilien PP80x und PP40x bietet dem PSi- Vertriebspartner die Möglichkeit, eigenständig Lösungen für besondere Anforderungen seines Kunden zu erarbeiten und seine Investitionen auch abzusichern. StringChange ist in allgemeiner Form die Behandlung von ankommenden Zeichenketten, die in modifizierter Form zum Drucken ausgegeben oder ganz eliminiert werden. Gerade bei Ablösungen von veralteten Drucklösungen muss auf Besonderheiten bestehender Applikationen eingegangen werden. Mit der Funktion StringChange können Befehle in andere umcodiert, vorhandene Befehle erweitert oder unerwünschte Steuerzeichen eliminiert werden. Voraussetzung ist natürlich die Eindeutigkeit der betroffenen Kodierungen und Funktionen. Ein gutes Verständnis im Umgang mit Druckerfunktionen und deren Kodierungen erleichtert die Arbeit.

1.1 Personality Modules mit der Funktion StringChange

Bestellnummer:

8707-241-90110
8707-241-90111
8707-241-90114

PP 40x:

PM PAR STRING CHANGE PP40x
PM ETH STRING CHANGE PP40x
PM ETH/USB PjL STGCHG PP40x

Bestellnummer:

8707-340-90127
8707-340-90128
8707-340-90142
8707-340-90143
8707-340-90144
8707-340-90145

PP 80x:

PM PAR STGCHG 80X MT50
PM ETH 10/100 STGCHG 80X MT50
PM ETH 10/100 MB/S PjL PP80x
PM ETH 10/100 MB/S PjL STGCHG PP80x
PM ETH 10/100 MB/S PjL IGP PP80x
PM ETH 10/100 MB/S PjL STGCHG IGP PP80x

2.1 Installation des PM's

Das Personality Module wird in den dafür vorgesehenen Schacht in den ausgeschalteten Drucker eingeschoben. Darauf achten, dass das Personality Module vollständig eingeschoben ist und die beiden Befestigungsschrauben von Hand gut angezogen sind. Die Schnittstellenkabel nur bei ausgeschaltetem Drucker einstecken.

3.1 Stecker, Taster und LED- Anzeige

Grüne LED Taster

Ethernet oder Parallelstecker



Zum Laden einer StringChange- Funktion (StringChange Definition)Taster gedrückt halten und Drucker einschalten und Taster wieder loslassen wenn die grüne LED leuchtet. Dann den Download beginnen. Bei längeren Download- Daten flackert die grüne LED. Der Download ist beendet, wenn die grüne LED nicht mehr leuchtet. Die Beschreibung zum Ethernet befindet sich in der Dokumentation zum Ethernet.



PP 80x



PP 404/5



PP 407/8



2 Anwendung und Installation

Im Vorfeld ist eine Analyse der erforderlichen Maßnahmen sinnvoll. Dies ist auch ohne Einsatz eines StringChange- Personality Moduls möglich. Es müssen lediglich die StringChange- Tools auf einem PC installiert werden und die Datenströme der Applikation müssen bekannt sein. Idealerweise sollten die Datenströme der Applikation als Datei aufgezeichnet sein. Nachdem die erforderliche Datenumsetzung analysiert ist, kann schon mit den StringChange Tools untersucht werden, wie die Umsetzung erfolgen könnte und ob das gewünschte Ergebnis erreicht werden kann. Dazu werden zwei Dateien, die StringChange- Definitionen und die Datei mit den Applikationsdaten mittels einem speziellen Programm umgewandelt und in einer Ausgabedatei abgespeichert. Diese Ausgabedatei kann mit einem sogenannten Hex-Editor untersucht werden und man kann sie auch direkt mit einem Standarddrucker ohne StringChange- Funktion ausdrucken, um das Druckergebnis zu prüfen.

2.1 Programmierung

2.1.1 Installation der StringChange Werkzeuge

Für die Sprachen Deutsch und Englisch sind eigene Verzeichnisse angelegt. Kopieren Sie die Verzeichnisse der gewünschten Sprache (Englisch oder Deutsch) von der CD auf den Rechner (PC) in ein neues Verzeichnis.

Kopieren Sie dann die DLL- Dateien aus dem Unterverzeichnis SYSTEM32 in das WINDOWS – Verzeichnis WINDOWS\SYSTEM32 und WINDOWS\SYSTEM

In den Verzeichnissen sind standardisierte Dateien angelegt, die die Erstellung von StringChange- Funktionen erleichtern sollen:

- **STRCHG_D.ini** - Datei für die Definition der StringChange Funktionen TESTSTG.bat BAT- Datei zur Überprüfung der STRCHG_D.ini auf Fehler und Kontrolle der Umsetzung.
- **mkSTGCHG.bat BAT** - Datei zur Erzeugung einer MOT- Datei zum Download in das StringChange Modul (PM)
- **STGCHG0000.mot**- Deaktiviert das StringChange Modul auf dem Personality Modul (StringChange ohne IGP)
- **Emu0000.mot** - Deaktiviert das StringChange Modul auf dem Personality Modul mit IGP- Funktion
- **Print_STGCHG_ver** - Druckt den Versions- Output der StringChange- Funktion

Hilfreich ist die Installation eines Hex.- Editors, mit dem die Applikationsdaten und die Testergebnisse untersucht werden können, bevor die StringChange Funktionen in einem Drucker installiert werden.

Ein kostenloser Hex- Editor WinVi kann von www.winvi.de heruntergeladen werden

3 Definition der StringChange-Funktionen

Eine StringChange Funktion erlaubt die Definition von Ersetzungsfunktionen durch Umcodieren eines eingehenden Datenstroms in einen anderen ausgehenden Datenstrom. Das sind in der Regel Steuerbefehle, die in andere Steuerbefehle umgewandelt werden müssen oder die Entfernung unerwünschter Zeichen aus dem Datenstrom zum Drucker. Variable Daten aus dem eingehenden Datenstrom können in den ausgehenden Datenstrom übernommen aber nicht umgewandelt werden. Analysieren Sie den Datenstrom der Applikation sorgfältig. Formulieren Sie eindeutige Gleichungen, die die umzuwandelnden Kodierungen des eingehenden Datenstroms mit der neuen Funktion in dem ausgehenden Datenstrom gegenüberstellen. Besondere Aufmerksamkeit erfordert die Handhabung von Variablen. Variable können Zeichenketten bestimmter oder variabler Länge mit eindeutigen Ende-kriterium oder auch Parameter (Werte) in einem Steuerbefehl sein.

In den ausgehenden Datenstrom können die variablen Daten an vordefinierte Stellen im Ausgabe-String übernommen oder auch eliminiert werden aber nicht in andere Werte umgewandelt werden. Daher ist eine gute Kenntnis der im Drucker vorhandenen Befehle und Steuerfunktionen von Vorteil, um die Ersetzungsfunktionen optimal zu definieren.

Generieren Sie sich eine der Applikation entsprechende Binärdatei. Diese kann von einer Aufzeichnung des Datenstroms einer Applikation stammen oder es werden selektiv die umzuwandelnden Teile aus einem HEX-DUMP in eine neue Datei editiert. Benennen Sie diese Datei für die Projektentwicklung mit STRCHG.inp. Verwenden Sie die vorhandene Datei STRCHG_D.ini um dort Ihre StringChange-Funktionen zu definieren.

In der Datei STRGCHG_D.ini sind einige Beispiele enthalten, die den Einstieg in die StringChange-Programmierung erleichtern sollen. Löschen Sie die Beispiele Trg ... / Out... und ersetzen Sie diese durch Ihre eigenen Definitionen. Sie können die Beispiele auch an das Ende der Definitionen in Kommentar mit einem #-Zeichen am Zeilenanfang setzen.

Gehen Sie bei der Entwicklung der StringChange-Definitionen schrittweise vor und überprüfen Sie immer wieder zwischendurch die korrekte Funktion durch Anwendung der Testfunktion TESTSTG.bat. TESTSTG.bat benötigt als Input die Dateien STRCHG_D.ini mit den StringChange-Definitionen und die Datei STRCHG.inp als Applikationsdatei. TESTSTG.bat generiert eine STGCHG.out-Datei mit den Daten, die durch die StringChange-Definitionen der STRCHG_D.ini nach Umsetzung aus der STRCHG.inp entstanden sind. Mit einem üblichen Hex-Editor kann das gewünschte Ergebnis der Umsetzung in der Datei STGCHG.out überprüft werden. Je nach Komplexität sind mehrere Durchläufe erforderlich, bis die gewünschte StringChange-Funktionen für die Anwendung eingesetzt werden kann.

Die STGCHG.out-Datei können Sie auch direkt zum Drucker übertragen, um den korrekten Druck der Applikation nach Umsetzung durch StringChange zu überprüfen. Dazu ist kein Personality Module (PM) mit StringChange-Funktion erforderlich. Es ist weniger zeitaufwändig, zunächst nur mit der Druckausgabe der Datei STGCHG.out zu testen, bevor dann die endgültige neue StringChange-Funktion im Drucker durch Download installiert wird. Diese Vorgehensweise ist auch dann interessant, wenn eine StringChange-Funktion erstellt oder geändert werden muss, ohne dass ein entsprechendes PM mit einem StringChange-Modul vor Ort zur Verfügung steht.

3.1 StringChange – Funktionen Mode 0 bis Mode 8

Neun unterschiedliche Modi stehen als StringChange- Funktion zur Verfügung:

Mode 0:

Zeichenketten suchen und ersetzen. Variable im Trigger möglich mit einem Zeichen pro Joker- Zeichen. Keine Übernahme der Variablen in den Output.

Mode 1:

Zeichenketten mit und ohne Joker suchen und mit „bistabiler“ Ausgabe ohne Übernahme von variablen Zeichen aus dem Trigger ersetzen. Beim ersten Auftreten der gesuchten Zeichenkette wird der erste definierte Output ausgegeben und beim nächsten Auftreten der zweite, dann wieder der erste Output usw.

Mode 2:

Zeichenketten mit variablen Zeichen variabler Länge suchen und ersetzen ohne Übernahme der variablen Zeichen. Die Aufnahme der variablen Daten erfolgt bis zum Zeichen nach dem Joker oder wenn das erste Zeichen des Triggers wiedergefunden wird.

Mode 3:

Suchen einer Zeichenkette, bei der das der Zeichenkette vorangestellte Zeichen in der Ausgabe an der durch das Übernahmezeichen gekennzeichneten Position so oft wiederholt wird, wie es dem Wert der durch Joker gekennzeichneten Variablen entspricht.

Mode 4:

Eine Zeichenkette, auch mit variablen Zeichen, soll entfernt werden.

Mode 5:

Erweiterungsfunktion zu Mode 1. Es können weitere Paare bi- stabile Ausgabe-Strings (Out) im Mode 1 zugewiesen werden.

Mode 6:

Counter, n- mal den Output Zerlegt einen großen Zählwert in kleinere Werte im Ausgabe- String.

Mode 7:

Suchen einer Zeichenkette mit variablen Daten, variabler Länge und Übernahme an einer definierten Position in der Ausgabe. Die Übernahme der Variablen ist optional.

Mode 8:

Suchen einer Zeichenkette mit einem oder mehreren variablen Zeichen. Die variablen Zeichen können in die Ausgabe übernommen werden Ein Übernahmezeichen repräsentiert in der Ausgabe ein variables Zeichen.

3.2 Format einer StringChange – Definition:

Für jede Ersetzungsfunktion (Trigger) muss der passende Mode gewählt werden. Die allgemeine Definition hat folgendes Format:

Den Trigger definieren: Trg..... und die Ausgabe definieren: Out.....

Trg <#> : <mode> , "<Definition>"

Out <#> : "<Definition>"

<#> : Fortlaufende Nummer der Trigger und Output- Definitionen, die aber nicht in aufsteigender Reihenfolge geordnet sein müssen.

- <mode> : Ziffer, die den Mode kennzeichnet : 0 bis 8
- < Definition> : Definition des Trigger- Strings bzw. Ausgabe- Strings.

Zwischen der Definition Trg ... und Out ... dürfen keine Leerzeilen eingegeben werden. Es gibt unterschiedliche Arten der Zeicheneingabe.

Normale ASCII Zeichen und hexadezimale Darstellung, z.B. ^0A für Line Feed, können verwendet werden. Die hexadezimale Darstellung ist immer dann notwendig, wenn das Zeichen nicht im Standard- ASCII Zeichensatz enthalten ist, es sich um Sonderzeichen oder nationale Sonderzeichen oder Steuerzeichen handelt. Bei überlangen Zeilenlängen der Definitionen, z.B. > 80 Stellen, wird die Zeile unterteilt, damit sie lesbar bleibt: "\" steht für Zeilenumbruch.

"<Definition>"\

"<Definition>"\

"<Definition>"\

"<Definition>"

Maximal darf eine Definition (Trigger oder Outstring) 255 Zeichen beinhalten. Werden für den Outstring mehr Zeichen angegeben, dann erscheinen in der Ausgabe FF Zeichen (hexadezimal), die Ausgabe ist fehlerhaft.

3.3 Anwendung von Jokern und Übernahmezeichen

Bei variablen Zeichen und Zeichenketten werden diese in der Trigger- Definition (Trg) mit einem Joker-Zeichen (?) und in der Ausgabe (Out) mit einem Übernahme-Zeichen (!) gekennzeichnet. Für einen Joker können beliebig viele Zeichen, begrenzt durch die Größe des Input – Buffers, aufgenommen werden mit Ausnahme im Mode 0, bei dem nur ein Zeichen pro Joker erlaubt ist.

Die Aufnahme von Zeichen durch Joker wird beendet durch:

- a) Das Zeichen nach dem Joker im Trigger
- b) Joker break - Character (!=\$00)
- c) Das erste Zeichen des Trigger wieder gefunden wird. Stehen aber 2 Joker hintereinander (??), so wird die Aufnahme von Zeichen nicht abgebrochen. Es können Joker 1 und Joker 2 definiert werden. Beide Joker sind in der Datenaufnahme gleichwertig. Der Unterschied besteht darin, dass nur Daten des Joker 1 in den Output übernommen werden können.

Für die Zeichenaufnahme mit Joker ist es nicht erforderlich, mehr als 2 Joker hintereinander im Trigger zu verwenden, es können aber beliebig viele Stellen mit Joker definiert werden.

Wird ein Joker als letztes Zeichen in einem Trigger angegeben, dann wird danach der Trigger beendet und nur ein Zeichen als Variable berücksichtigt. (Beispiel hierzu in der Beschreibung zu MODE 8)

3.4 Welcher Trigger wird wirksam?

Generell wird derjenige Trigger wirksam, der die meisten Zeichen aufgenommen hat. Sind mehrere Trigger mit den gleichen Daten und Datenmengen aktiv und haben einen Match, dann ist derjenige gültig, der zuerst im Init- File angelegt wurde.

Werden gleiche Trigger definiert, die sich lediglich durch feste und variable Werte unterscheiden, dann sollen die Trigger mit den festen Werten vor den Triggern mit variablen Werten im Init- File angelegt werden.

4 Beschreibungen und Beispiele der einzelnen Modi

Die einfachen Beispiele sollen das Prinzip und die Wirkungsweise der verschiedenen Modi verdeutlichen. Die komplexeren Beispiele sind eher aus der Praxis.

4.1 Mode 0

Eine Zeichenkette (Trigger) wird durch eine andere Zeichenkette (Output) ersetzt. Im Trigger können Joker für Variable verwendet werden. Die Zeichen aus den Joker-Positionen können aber nicht in den Ausgabe-String übernommen werden. Ein Joker steht für jeweils ein variables Zeichen.

Beispiel 1:

Ein einfaches Beispiel ist die Ersetzung eines Namens (aus PP405 soll PP806 werden, aber alle anderen Varianten werden PP809):

Trigger 1:0

```
Trg 1: 0, "PP405"  
Out 1: "PP806"
```

Wird die Zeichenfolge PP405 im Datenstrom gefunden, dann wird diese mit der Zeichenfolge PP806 ersetzt.

Die Zeichenfolge PP mit 3 weiteren variablen Zeichen soll in die Zeichenfolge PP809 umgewandelt werden. Ausnahme ist die Zeichenfolge PP405, die als feste Zeichenfolge mit Trigger 1 vor Trigger 2 definiert ist und PP806 als Zeichenfolge ausgibt. Weil im Mode 0 ein Joker nur ein variables Zeichen definiert, sind hier in der Trigger- Definition zu Trg 2 für drei variable Zeichen auch drei Joker angegeben.

Trigger 2:0

```
Trg 2: 0, "PP???"  
Out 2: "PP809"
```

Beispiel 2:

Ein Beispiel aus der Praxis:

Es soll eine Barcode – Definition durch einen entsprechenden PSi Barcode ersetzt werden.

Trigger 3, 4 und 5:0

Trg 3: 0,"^14^14^1b^21^02^11^00" # Barcodeheader ersetzen

Out 3: " ^1b^5b;103;;;;; z"

Trg 4: 0,"^19^14^14^1b^28^1d^00" # Start Barcode ersetzen

Out 4: " ^1b^5b?0h:"

Trg 5: 0,"^19^14^14^1b^59^02" # Stop Barcode ersetzen

Out 5: " ;^1b^5b?0l"

Wird die Zeichenfolge des Triggers (Trg 3/4/5) im Datenstrom gefunden, dann wird diese durch die Zeichenfolge (Out 3/4/5) jeweils ersetzt.

In diesem Fall muss die Geometrie des Barcodes ermittelt und die entsprechenden Werte in den Barcode-Header der Ausgabe (Out 3) eingesetzt werden. Konkret sind das Code 39, Barcode- Höhe 5/12 ", Breite der schmalen Striche 2/144 ", Breite der schmalen Lücken 2/144 ", Verhältnis breit zu schmal 3.0 : 1. Eventuelle START/STOP Sequenzen müssen ebenfalls gesucht und ersetzt werden.

4.2 Mode 1

Für einen Trigger wird eine bi- stabile Ausgabe (2 Output- Strings) definiert. Beim ersten Auftreten der Trigger- Definition wird die erste Ausgabezeichenkette, bei dem nächsten Trigger die zweite Ausgabezeichenkette (Out 1.1) und dann wieder die erste Ausgabezeichenkette ausgegeben usw. Als reale Anwendung käme z. B. eine Ein- und Ausschaltfunktion in Betracht, z. B. „Fettdruck ein“ und „Fettdruck aus“

Beispiel 1:

Trigger 1:1

Trg 1: 1,"PSI"

Out 1: "Printers for professionals ^0D^0A"

Out 1.1: "PP806 ^0D^0A"

^0D^0A steht für die Steuerbefehle CR LF.

Input – Daten der Applikation:

PSIPSIPSIPSIPSIPSI

Nach Umwandlung der Input- Daten werden folgende Output- Strings ausgegeben:

Printers for professionals

PP806

Printers for professionals

PP806

Printers for professionals

PP806

Printers for professionals

PP806

Printers for professionals

PP806

Printers for professionals

PP806

In der Trigger- Definition können auch Joker verwendet werden.

Beispiel 2:

Trigger 2:1

Trg 2: 1, "?PSI"

Out 2: "! Printers for professionals ^OD^OA"

Out 2.1: "PP806 ^OD^OA"

^OD^OA steht für die Steuerbefehle CR LF.

Input – Daten der Applikation:

1PSI2PSI3PSI4PSI5PSI6PSI

Nach Umwandlung der Input- Daten werden folgende Output- Strings ausgegeben:

1 Printers for professionals

PP806

2 Printers for professionals

PP806

3 Printers for professionals

PP806

4 Printers for professionals

PP806

5 Printers for professionals

PP806

6 Printers for professionals

PP806

4.3 Mode 2

Mode 2 definiert die Ersetzungsfunktion für einen String mit variablen Daten ohne deren Übernahme in den Output- String. Die Aufnahme der variablen Daten erfolgt bis zum Zeichen nach dem Joker oder wenn das erste Zeichen des Triggers wieder gefunden wird. Die zweite Bedingung gilt nur, wenn im Trigger ein Joker-Zeichen (?) statt zwei Joker-Zeichen (??) angegeben werden.

Beispiel 1:

Ein Trigger mit variablen Zeichen (Joker) wird definiert. Die Ausgabe soll ohne Übernahme der variablen Zeichen eine feste Zeichenkette sein. Im Gegensatz zu Mode 0 werden ein oder mehrere Zeichen bis zum Erreichen des Zeichens nach dem Joker aufgenommen.

In der INI- Datei wird folgender Trigger definiert:

Trigger 1:2

Trg 1: 2, "P??!"

Out 1: "PP803"

In den Input- Daten werden folgende Zeichenketten als Trigger erkannt:

PSI PPI PTI PLI P1I PXI PxxxxI PyyyyyyyyyyI P123467890I

Und werden nach Umwandlung jeweils als feste Zeichenkette in den Output gestellt:

PP803 PP803 PP803 PP803 PP803 PP803 PP803 PP803 PP803 PP803

Beispiel 2:

Ein Beispiel aus der Praxis:

Ein String beginnt immer mit einem oder mehreren bekannten Zeichen, gefolgt von einer festen Anzahl variabler Zeichen (Joker) und endet immer mit einem oder mehreren bekannten Zeichen. Es handelt sich hier um unerwünschte Zeichen im Datenstrom, die eliminiert werden müssen.

Die Definition sollte möglichst komplex definiert werden, damit nicht andere Daten der Applikation ungewollt einbezogen werden.

Trigger 1:2

Trg 1: 2,“^0A^00??????^20“

Out 1: “^0A^20“

Alle Daten zwischen ^0A und ^20 werden entfernt. Hier wurde ^00 mit in die Definition einbezogen, damit die Definition eindeutiger wird und nicht auch andere Daten ungewollt entfernt werden.

4.4 Mode 3

Mode 3 wiederholt das Zeichen vor dem Trigger mit dem Wert in der durch Joker “?” markierten Position in der Ausgabe an der durch “!” gekennzeichneten Position.

Die Variable “?” hat Default den Wert 1 und wird auf maximal 100 in der Ausgabe begrenzt. Der Wert für die Variable wird mit ^30 bis ^39 (hexadezimal) kodiert.

Beispiel 1:

Wird der Trigger im Eingangsdatenstrom erkannt, dann soll das Zeichen vor dem Trigger so oft ausgegeben werden, wie in der Position mit dem Joker (?) definiert ist.

Definition in der INI- Datei:

Trigger 1:3

Trg 1: 3,“PSI?test“

Out 1: “!”

Input – Daten der Applikation: ^PSI4test

Output nach Umwandlung: ^^^^

Hier befindet sich das Zeichen ^ vor dem Trigger im Input- Datenstrom und wird 4-mal ausgegeben.

Beispiel 2:

Der ANSI- Befehl Repeat Charakter wird durch die Ausgabe einer entsprechenden Anzahl Zeichen ersetzt.

Trigger 1:3

Trg 1: 3,“^1B^5B?b“

Out 1: “!”

In den Ausgabe- String können auch Zeichen vor und nach dem Übernahmezeichen “!” eingesetzt werden.

4.5 Mode 4

Mode 4 wird als Filter verwendet. Der Trigger wird aus dem Ausgabedatenstrom entfernt.

Beispiel 1:

Die Zeichenkette "OLD" soll aus dem Datenstrom entfernt werden.

Definition in der INI- Datei

Trigger 1:4

Trg 1: 4, "^22OLD^22"

^22 repräsentiert das Zeichen "

Input Daten:

"OLD"

Beispiel 2:

Eine Zeichenkette mit variablen Zeichen soll entfernt werden.

Trigger 2:4

Trg 2: 4, "alte Ausgabe ?? "

Das letzte Zeichen im Trigger ist ein Leerzeichen (Space).

Input Daten:

Test 1 alte Ausgabe 1 !

Test 2 Das ist eine alte Ausgabe von 1999

Test 3 diese alte Ausgabe xxxxxxxxyyyyyzzzzzz ?

Output Daten:

Test 1 !

Test 2 Das ist eine 1999

Test 3 diese ?

4.6 Mode 5

BISTABIL Name, legt Name des Output fest.

Erweiterungsfunktion zu Mode 1. Es können weitere Paare bi- stabile Ausgabe-Strings (Out) mit Namen im Mode 1 zugewiesen werden. Der Name ist 1 Zeichen lang und wird von der Position des Jokers gelesen. In der Ausgabe (Out) wird der Name (1 Zeichen) in der ersten Position angegeben. Der Name wird nicht ausgegeben. Nach dem letzten Output String muss immer eine Leerzeile stehen. In der Applikation soll nach der letzten Output- String ein \$FF stehen. Ist kein Name angegeben oder ein Name ist nicht definiert, dann wird das erste

Ausgabepaar ohne Namen (1.1 und 1.2) verwendet.

Beispiel :

Trigger 1:5

```
Trg 1: 5, "Toggle_"  
Out 1.1: "TOGGLE ON "  
Out 1.2: "TOGGLE OFF "  
Out 1.3: "xTOGGLE X "  
Out 1.4: "yTOGGLE Y "  
Out 1.5: "zTOGGLE Z "  
Trg 2: 5, "Bistab?Name"  
Out 2: "!"
```

Zur Übernahme des Namens, der durch Mode 5 definiert wird, ist mindestens das Übernahmezeichen "!" im OUTPUT String zu definieren.

Input Daten:

```
a Toggle_  
b Toggle_  
a Toggle_  
b Toggle_  
BistabxName  
c Toggle_  
d Toggle_  
c Toggle_  
d Toggle_  
BistabyName  
e Toggle_  
f Toggle_  
e Toggle_  
f Toggle_
```

Output Daten:

```
a TOGGLE ON  
b TOGGLE OFF  
a TOGGLE ON  
b TOGGLE OFF  
c TOGGLE X  
d TOGGLE Y  
c TOGGLE X  
d TOGGLE Y  
e TOGGLE Y  
f TOGGLE Z  
e TOGGLE Y  
f TOGGLE Z
```

4.7 Mode 6

Counter, n- mal den Output

Zerlegt einen größeren Zählwert in kleinere Werte im Ausgabe- String. Werte größer 500 werden in der Ausgabe in $n * 500$ plus Restwert zerlegt. Die Anwendung dieser Funktion ist sehr speziell.

Beispiel zur Verdeutlichung der Wirkungsweise

Beispiel :

Trigger 1:6

Trg 1: 6, "Count ? Mode"

Out 1: "COUNT ! times"

Input Daten:

Count 1 Mode
Count 20 Mode
Count 400 Mode
Count 1531 Mode

Output Daten:

COUNT 1 times
COUNT 20 times
COUNT 400 times
COUNT 500 times
COUNT 500 times
COUNT 500 times
COUNT 31 times

4.8 Mode 7

Mode 7 definiert eine Ersetzungsfunktion mit variablen Daten, variabler Länge. Die Variable im Trigger wird mit einem "?" (Joker) bezeichnet In der Ausgabe (Out) wird die Variable aus dem Trigger in die Position eingesetzt, die mit einem Übernahmezeichen "!" bezeichnet ist. Es kann auch eine Ausgabe (Out) ohne Übernahme einer Variablen definiert werden.

Beispiel:

"^2A?^2A^19" ist ein Teil eines Barcodes, die Variablen Zeichen des Barcodes sind mit ^2A eingeklammert. Es handelt sich hier um den Code 39.

Die Ausgabe soll einen Barcode nach PSI- Syntax erzeugen mit Übernahme der variablen Daten.

"^1B^5B^3F^30^68^2A!^2A^2A^1B^5B^3F^30^6C^0A^0A^0A"

Start Barcode : ^1B^5B^3F^30^68

Variable Barcode Daten übernehmen: ^2A!^2A^2A

Das „!“ markiert als Übernahmezeichen die Position der Variablen im Ausgabestring, ^2A muss vor und nach der variablen wieder eingesetzt werden, da sie als Teil des Triggers verloren gegangen sind und ein weiteres ^2A ist notwendig, weil ein ^2A das Ende der Variablen für die StringChange- Funktion kennzeichnet. Stopp Barcode : ^1B^5B^3F^30^6C als Abschluss der variablen Barcode- Daten. Vertikaler Vorschub: ^0A^0A^0A, damit die folgenden Daten in die richtige Zeile gedruckt werden. Gerade bei Barcode ist die aktive Druckposition je nach Emulation unterschiedlich definiert und wird hier mit ^0A ausgeglichen.

Vollständige Ersetzungsfunktion:

Trigger 1:7

Trg 1: 7, "^2A?^2A^19"

Out 1: "^1B^5B^3F^30^68^2A!^2A^2A^1B^5B^3F^30^6C^0A^0A^0A"

Beispiel:

Beispiel für das Filtern eines Daten-Strings mit variablen Daten. Die variablen Daten müssen eindeutig mit bekannten Zeichen eingeklammert sein. Das folgende Beispiel enthält variable Daten mit einem eindeutig definierten Anfang und Ende: CR, LF vor und Space nach den variablen Daten. Die variablen Daten sollen eliminiert werden.

Trg 1:7 "^0D^0A?^20"

Out 1: "^0D^0A^20"

4.9 Mode 8

Mode 8 definiert die Übernahme eines oder mehrerer variablen Zeichen mit einem Zeichen pro Übernahmezeichen.

Im INIT- File werden zwei Trigger für MODE 8 definiert.

Trigger 1 soll die ersten vier variablen Zeichen nach <ESC> (in die Ausgabe (Out) übernehmen). Der Trigger ist aber nur gültig, wenn nach den variablen Daten wieder ein <ESC> gefunden wird.

Trigger 1:8

Trg 1: 8, "^1B(????^1B"

Out 1: "!!!!"

Trigger 2 soll die ersten drei variablen Zeichen nach einem <ESC> in die Ausgabe übernehmen gefolgt von den festen Zeichen Drucker. Der Trigger ist auch hier nur gültig, wenn nach den variablen Daten ein <ESC> gefunden wird.

Trigger 2:8

Trg 2: 8, "^1B???????^1B"

Out 2: "!!! Drucker"

Im Input- File sind folgende Daten enthalten:

<ESC>(TESTTESTTESTTESTTESTTESTTEST<ESC>

<ESC>PSIPSIPSI<ESC>

<ESC> steht für das Zeichen Escape mit dem Code hex 1B.

Als Ausgabe (Output- File) werden nach Umwandlung durch die Definitionen im INITFile

TEST

PSI Drucker

ausgegeben.


```

#JOKERBREAK: dieses Zeichen im Datenstrom beendet einen Trigger, wenn Joker aktiv ist
# ^00=inaktiv, Range: ^00..^FF
#UEBERNAHME: dieses Zeichen im Output: es werden Zeichen aus den
Eingangsdaten
# ab der Position des Joker 1 im Trigger uebernommen, Range: ^00..^FF
HEX_ESC ^^ # Binary-Init-File: ^0B[^5E], default ^5E
LINE_ESC \ # Binary-Init-File: ^0C[^5C], default ^5C
UDE_CHAR ^^ # Binary-Init-File: ^09[^5E], default ^5E
UDE_MODE D # Binary-Init-File: ^08[^44], default ^44
JOKER1 ? # Binary-Init-File: ^04[^3F], default ^3F
JOKER2 ^00 # Binary-Init-File: ^05[^00], default ^00
JOKERBREAK ^00 # Binary-Init-File: ^06[^00], default ^00
UEBERNAHME ! # Binary-Init-File: ^07[^21], default ^21
NOT_UDE "^^CR"
NOT_UDE "^^EX"
NOT_UDE "^^AB"
NOT_UDE "^^34"

```

Beispiele :

```

Trg 1: 0, "PSi"
Out 1: "Printers for professionals PP809, PP806, PP803 "
Trg 2: 0, ^50^65^72^73^6F^6E^61^6C^69^74^79^20^4D^6F^64^75^6C"
Out 2: "PM"
Trg 3: 0, ^0D^0ATE^53^54"
Out 3: ^0D^0AO^4B"
Trg 10: 1, "PSi"
Out 10: "Excellence in printing solutions^0D^0A"
Out 10.1: "Creating printers for professionals^0D^0A"
Trg 20: 2, "P?!"
Out 20: "PP803"
Trg 30: 3, "PSI?test"
Out 30: "!"

```

usw.....

Die Beispiele Trg/Out können gelöscht oder als Kommentarzeilen an das Ende der STGCHG.ini- Datei verschoben werden.

Vor den StringChange- Definitionen (Trg/Out)) darf keine Kommentarzeile (#) eingegeben werden. Zwischen den Zeilen Trg und Out dürfen keine Leerzeilen eingegeben werden. Die Datei wird nur mit einem Standard- ASCII- Zeichen Editor bearbeitet.

5.1.1 StringChange- Definitionen testen

Nachdem die Datei STGCHG_D.ini mit den gewünschten Definitionen fertiggestellt ist, und eine Datei mit dem Eingangsdatenstrom der Applikation vorliegt (STGCHG.inp) kann der erste Test einer Umsetzung in die Ausgabedatei STGCHG.out erfolgen:

Beide Dateien befinden sich in einem Ordner mit der Datei TESTSTG.bat. TESTSTG.bat starten und die Ausgabedatei STGCHG.out generieren.

Die Abbildung zeigt den Inhalt einer TESTSTG.bat – Datei.

```

Datei Bearbeiten Suchen ?
REM Test fuer die StringChange Definitionen:
REM STGCHG_D.ini: Definitionen fuer das StringChange Interface Modul
REM STGCHG.inp  : INPUT Definitionen = Test Datei oder Applikationsdatei (binaer)
REM STGCHG.out  : OUTPUT StringChange (Datei)
REM Datei STGCHG.out dient zur Ueberpruefung und Fehlersuche (DEBUGGING)
REM der StringChange Definitionen in der Datei STGCHG_D.ini

msc_aust -b1 STGCHG_D.ini STGCHG.inp STGCHG.out

```

Die Abbildung zeigt ein DOS- Fenster nach einem erfolgreichen Durchlauf von TESTSTG.bat mit einer kleinen STGCHG_D.ini – Datei.

```

H:\Rv0\STRGCHGE\TEST>REM COMPARE THE STGCHG.out WITH YOUR DEFINITIONS FOR DEBUG
ING
H:\Rv0\STRGCHGE\TEST>
H:\Rv0\STRGCHGE\TEST>REM msc_aust -blev STGCHG.ini STGCHG.inp STGCHG.out
H:\Rv0\STRGCHGE\TEST>msc_aust -b1 STGCHG.ini STGCHG.inp STGCHG.out

alloc memory: 472 Byte at $00770900
fixed size: 424 Byte
variable size: 2 Trg/Out * 24 Byte = 48 Byte

//Match Trigger 1, Tg-Len=5 Act-Tg-Len=5//
//Match Trigger 2, Tg-Len=5 Act-Tg-Len=5//
//Match Trigger 2, Tg-Len=5 Act-Tg-Len=5//
//Match Trigger 2, Tg-Len=5 Act-Tg-Len=5//
H:\Rv0\STRGCHGE\TEST>
H:\Rv0\STRGCHGE\TEST>
H:\Rv0\STRGCHGE\TEST>
H:\Rv0\STRGCHGE\TEST>
H:\Rv0\STRGCHGE\TEST>

```

belegter Speicherplatz

alloc memory : = belegter Speicherplatz

Mit einem HEX- EDITOR kann nun die Umsetzung der StringChange- Definitionen in der Datei STGCHG.out überprüft werden.

Diese Abbildung zeigt den Inhalt einer STGCHG.out – Datei mit Hilfe eines HEX-Editors.

```

te.prm* - WinVi
Datei Bearbeiten Suchen Optionen Fenster Hilfe
[Icons] UTF UTF 8 16 89
000000: 50 50 38 30 33 20 50 50 38 30 36 20 50 50 |PP803 PP806 PP80|
000016: 39 |9|
~
~
~

```

Zusätzlich kann die Datei STGCHG.out zum Drucker übertragen werden, um den korrekten Ausdruck zu überprüfen. Vorher muss auf die korrekte Menüeinstellung des Druckers hinsichtlich Emulation usw. geachtet werden. Entspricht das Ergebnis nicht den Erwartungen, dann sind die StringChange-Definitionen in der Datei STGCHG_D.ini zu überprüfen und zu korrigieren.

5.1.2 StringChange- Definitionen in das Personality Modul laden

Nachdem die StringChange- Definitionen auf korrekte Funktion überprüft wurden, kann nun der Download in das Personality Modul erfolgen. Die StringChange Funktionen werden mit einer Datei STGCHG.mot in das Personality Modul geladen. Zur Erstellung dieser Datei wird mkSTGCHG.bat gestartet. Die Datei STGCHG_D.ini muss sich im gleichen Verzeichnis befinden. Abhängig vom verwendeten Personality Modul muss vorher der korrekte Eintrag einer Adresse in der Datei mkSTGCHG.bin überprüft bzw. angepasst werden.

```

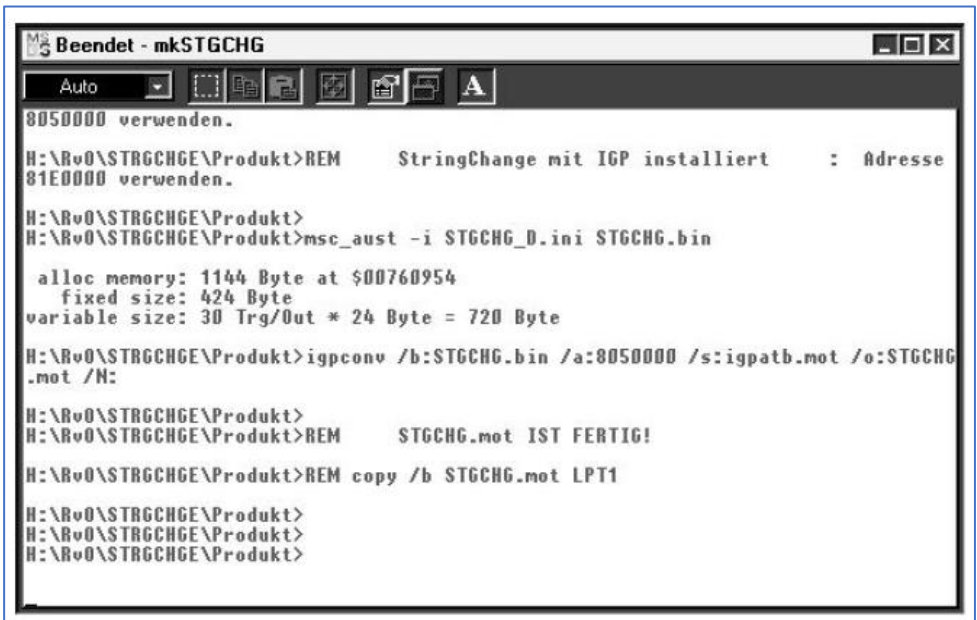
Datei Bearbeiten Suchen ?
REM      Version mkSTGCHG.bat 29.07.2005
REM      Generiert Datei STGCHG.mot zum Download in das StringChange- Interface Modul.
REM      Vor dem Download ist das StringChange Interface durch Einschalten mit gedruckter
REM      Taste fuer Download zu aktivieren.
REM      Definition der zu verwendenden Adresse:
REM      StringChange (ohne IGP) installiert : Adresse 8050000 verwenden.
REM      StringChange mit IGP installiert   : Adresse 81E0000 verwenden.

msc_aust i STGCHG_D.ini STGCHG.bin
igpconv /b:STGCHG.bin /a:8050000 /s:igpatb.mot /o:STGCHG.mot /N:

REM      STGCHG.mot IST FERTIG!
REM      copy /b STGCHG.mot LPT1

```

Die Abbildung zeigt ein DOS- Fenster nach erfolgreicher Umsetzung einer STGCHG_D.ini Datei in eine STGCHG.mot- Datei.



Die Datei STGCHG.mot kann jetzt z.B. mittels COPY- Befehl über die Parallelschnittstelle zum Drucker übertragen werden. Hierzu den Drucker ausschalten und mit gedrückter Taste () den Drucker wieder

einschalten. Die grüne LED neben dem Taster leuchtet. Anschließend Datei STGCHG.mot zum Drucker übertragen.

z.B. copy /b STGCHG.mot LPT1 bei Anschluss an eine Parallelschnittstelle oder bei Anschluss an eine Ethernet- Schnittstelle mittels FTP.

Hinweis:

Nach dem Download einer StringChange- Funktion kann das konfigurierte Personality Modul in einen anderen Drucker

5.1.3 StringChange- Definitionen zurücksetzen

Grundsätzlich kann mit keiner Funktion am Drucker (z.B. Tastendruck oder Menü) ein Ausdruck der installierten StringChange- Definitionen ausgelöst werden. Diese Maßnahme schützt vor unerwünschtem Zugriff auf die installierten StringChange- Funktionen.

Damit aber eine Kontrolle der installierten Funktionen möglich ist, befindet sich im Tool- Paket eine speziell kodierte Datei print_stgchg_ver.txt, die einen Ausdruck der Einstellungen bewirkt.

5.1.4 Hinweise zum Update der Drucker- Firmware

StringChange- Funktionen können den Firmware – Code verändern, wenn zufällig Trigger- Bedingungen im Code auftauchen und diese dann umgewandelt werden. Die Firmware ist dann in der Regel nicht lauffähig oder fehlerhaft. Bei einem Update der Drucker- Firmware kann man dieses Problem umgehen, wenn die PM- Basisplatine ausgebaut und ohne die obere IPG/ATB- Platine in den Drucker eingesetzt wird. Der Firmware – Update erfolgt dann über ein spezielles Parallel-Kabel an der internen Parallelschnittstelle der PM- Basisplatine. Alternativ kann man die StringChange Funktion vor dem Firmware Update ausschalten durch Anwendung der Datei STGCHG0000.mot. Nach erfolgtem Firmware- Update muss dann allerdings die StringChange Funktion wieder installiert werden.

5.3 Technische Daten

Anzahl von StringChange- Definitionen 32767 maximal

Anzahl Bytes pro Trigger (Input) – String 255 Bytes maximal

Anzahl Bytes pro Out (Output) – String 255 Bytes maximal

Anzahl Zeichen für Trigger / Output Paare 65520 Bytes

*) Alle Maximalwerte werden durch die tatsächlich verfügbare Speicherkapazität begrenzt. Praktisch kann man damit rechnen, dass die Anzahl von Trigger / Output Strings einer sehr komplexen Anwendung nicht über 1000 Definitionen liegt.