

# CPCL Font Manual

for ZQ210 / ZQ220 Printers



**ZEBRA**

ZEBRA and the stylized Zebra head are trademarks of Zebra Technologies Corporation, registered in many jurisdictions worldwide. All other trademarks are the property of their respective owners.  
© 2021 Zebra Technologies Corporation and/or its affiliates. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements.

For further information regarding legal and proprietary statements, please go to:

SOFTWARE: [zebra.com/linkoslegal](https://zebra.com/linkoslegal)

COPYRIGHTS: [zebra.com/copyright](https://zebra.com/copyright)

WARRANTY: [zebra.com/warranty](https://zebra.com/warranty)

END USER LICENSE AGREEMENT: [zebra.com/eula](https://zebra.com/eula)

## Terms of Use

### Proprietary Statement

This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries ("Zebra Technologies"). It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the express, written permission of Zebra Technologies.

### Product Improvements

Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

### Liability Disclaimer

Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

### Limitation of Liability

In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

## Publication Date

September 15, 2021

# Contents

<b>CPCL Fonts for Mobile Printers .....</b>	<b>4</b>
Built in CPCL Fonts.....	5
Font Encodings .....	6
Font Encodings – Single Byte (ASCII Encoding).....	6
USA or Dynamic – COUNTRY USA.....	6
United Kingdom – COUNTRY UK.....	6
German – COUNTRY GERMANY .....	7
French – COUNTRY FRANCE.....	7
Italian – COUNTRY ITALY .....	7
Swedish – COUNTRY SWEDEN .....	8
Spanish – COUNTRY SPANISH.....	8
Norwegian – COUNTRY NORWAY .....	8
LATIN 9 Encoding – COUNTRY LATIN9 .....	9
Code Page 850 Encoding – COUNTRY CP850.....	9
Code Page 874 Encoding – COUNTRY CP874.....	9
Font Encodings – Multi-byte.....	12
CHINA, JAPAN and KOREA Encodings .....	12
BIG5 Encoding .....	12
JAPAN-S (Shift-JIS) Encoding .....	13
THAI Encoding .....	13
VIETNAM Encoding.....	13
UTF-8 .....	13
GB18030 .....	13
Encoding Types for Common Fonts .....	14
Single-Byte Font Character List .....	15

# CPCL Fonts for Mobile Printers

This document focuses on the fonts available and/or pre-installed on the ZQ200 series mobile printers. For ESC/POS use of FontA/B/C, please refer to the ESC/POS ZQ210/ZQ220 Code Page Manual (part number P1124780-01EN).

In CPCL there are a number of different supported character encodings. The active encoding in CPCL is selected by the **CHAR-SET**, **COUNTRY** or **ENCODING** command.

```
CHAR-SET [Name]<CR><LF>  
COUNTRY [Name]<CR><LF>
```

<b>Command Name</b>	CHAR-SET, COUNTRY, ENCODING
<b>Short Form Aliases</b>	None
<b>Valid Session Types</b>	Label and Utilities Sessions
<b>Linked SGD</b>	None

This command is used to set the encoding for CPF and CSF fonts in CPCL. For more information about the various internal supported options, see the next section.

Each country code is given a name which is used with the **CHAR-SET**, **COUNTRY** or **ENCODING** command to activate it.

Available encodings are divided into two categories, single byte and multi-byte. Multi-byte fonts are covered in the next section.

## Built in CPCL Fonts

There are 7 built-in bitmap fonts in CPCL, with numbers from 0 to 7.



**NOTE:** There is no font 3.

The following table shows the font numbers as they would be used in the printer, followed by the size value, the width and height multipliers, and finally the character height and character width.

Font	Size	Width	Height	Character Height	Character Width
0	0	1	1	9	8
0	1	2	1	9	16
0	2	1	2	18	8
0	3	2	2	18	16
0	4	3	2	18	32
0	5	2	3	36	16
0	6	3	3	36	32
1	0	1	1	48	8-25 Variable
2	0	1	1	12	20
2	1	1	2	24	20
4 (A)	0	1	1	47	8-43 Variable
4 (A)	1	1	2	94	8-43 Variable
4 (B)	2	1	½	45	26-51 Variable
4 (B)	3	1	1	90	26-51 Variable
4 (B)	4	1	2	180	26-51 Variable
4 (B)	5	1	3	270	26-51 Variable
4 (B)	6	1	4	360	26-51 Variable
4 (B)	7	1	5	450	26-51 Variable
5	0	1	1	24	5-23 Variable
5	1	1	2	48	5-23 Variable
5	2	2	2	46	8-39 Variable
5	3	2	3	92	8-39 Variable
6	0	1	1	27	28
7	0	1	1	24	12
7	1	1	2	48	12

The table shows font 4 with both A and B options. This font is only referred to as font 4 but has two glyph sets.

## Font Encodings

### Font Encodings – Single Byte (ASCII Encoding)

In the single-byte encodings, there are tables which relocate characters in the font to make up the encoding. For these encodings, the character pages in the font must be defined using the following character placement. This encoding is Code Page 1252, with a couple of modifications, which are highlighted.

Not all characters are defined in all fonts, including the built in fonts.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0x0																
0x1																
0x2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0x3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0x4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0x5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0x6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0x7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
0x8	€		,	f	„	...	†	‡	^	‰	Š		Œ	š	Ž	Ÿ
0x9		‘	’	“	”	•	—	~	™	š	ø	œ	Œ	ž	ž	
0xa		ı	ç	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
0xb	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
0xc	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
0xd	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
0xe	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
0xf	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

The following encodings use this table by moving characters.

### USA or Dynamic – COUNTRY USA

The USA country code has no replacements and all characters pass through directly as defined in the font file. This can be used to replicate encodings which CPCL doesn't support by making a custom font.

### United Kingdom – COUNTRY UK

Substitute This Character	With This Character
0x23	#
0x7E	~
	0xA4
	£
	0xAF
	—

**German – COUNTRY GERMANY**

Substitute This Character		With This Character	
0x40	@	0xA7	§
0x5B	[	0xC4	Ä
0x5C	\	0xD6	Ö
0x5D	]	0xDC	Ü
0x7B	{	0xE4	ä
0x7C		0xF6	ö
0x7D	}	0xFC	ü
0x7E	~	0xDF	ß

**French – COUNTRY FRANCE**

Substitute This Character		With This Character	
0x23	#	0xA4	£
0x40	@	0xE0	à
0x5B	[	0xB0	°
0x5C	\	0xE7	ç
0x5D	]	0xA7	§
0x60	`	0xB5	µ
0x7B	{	0xE9	é
0x7C		0xF9	ù
0x7D	}	0xE8	è
0x7E	~	0xA8	¨

**Italian – COUNTRY ITALY**

Substitute This Character		With This Character	
0x23	#	0xA4	£
0x40	@	0xA7	§
0x5B	[	0xB0	°
0x5C	\	0xE7	ç
0x5D	]	0xE9	é
0x60	`	0xF9	ù
0x7B	{	0xE0	à
0x7C		0xF2	ò
0x7D	}	0xE8	è
0x7E	~	0xEC	ì

**Swedish – COUNTRY SWEDEN**

Substitute This Character		With This Character	
0x23	#	0xA3	¤
0x40	@	0xC9	É
0x5B	[	0xC4	Ä
0x5C	\	0xD6	Ö
0x5D	]	0xC5	Å
0x5E	^	0xDC	Ü
0x60	`	0xE9	é
0x7B	{	0xE4	ä
0x7C		0xF6	ö
0x7D	}	0xE5	å
0x7E	~	0xFC	ü

**Spanish – COUNTRY SPANISH**

Substitute This Character		With This Character	
0x23	#	0xA3	£
0x40	@	0xA7	§
0x5B	[	0xA1	¡
0x5C	\	0xD1	Ñ
0x5D	]	0xBF	¿
0x7B	{	0xB0	°
0x7C		0xF1	ñ
0x7D	}	0xE7	ç
0x7E	~	0xAF	—

**Norwegian – COUNTRY NORWAY**

Substitute This Character		With This Character	
0x5B	[	0xC6	Æ
0x5C	\	0xD8	Ø
0x5D	]	0xC5	Å
0x7B	{	0xE6	æ
0x7C		0xF8	ø
0x7D	}	0xE5	å
0x7E	~	0xAF	—



## LATIN 9 Encoding – COUNTRY LATIN9

Substitute This Character		With This Character	
0xA4	¤	0x80	€
0xA6	ı	0x8A	Š
0xA8	¨	0x8D	š
0xB4	’	0x8E	Ž
0xB8	‚	0x9F	ž
0xBC	¼	0x9D	Œ
0xBD	½	0x9C	œ
0xBE	¾	0x8F	Ÿ

## Code Page 850 Encoding – COUNTRY CP850

The font layout described in [Font Encodings – Single Byte \(ASCII Encoding\)](#) on page 6 does not contain all the necessary characters to fill out Code Page 850, in particular the line drawing and blocks.

As a result, any characters which are not available are replaced with the space character. The highlighted characters in the following code page 850 chart are not available in CPCL using the standard mappings.



**NOTE:** Please refer to the [Font Encodings – Multi-byte](#) on page 12 example to print these highlighted characters.


	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0x0																
0x1																
0x2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0x3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0x4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0x5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0x6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0x7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
0x8	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
0x9	É	æ	Æ	ô	ö	ò	ù	û	ÿ	Ö	Ü	ø	£	Ø	×	f
0xa	á	í	ó	ú	ñ	Ñ	ª	º	¿	®	¬	½	¼	¡	«	»
0xb	⋮	⋮	⋮		†	À	Â	À	©	¶	¶	¶	¶	¢	¥	¶
0xc	⋮	⋮	⋮	†	–	†	ä	Ä	⋮	⋮	⋮	⋮	⋮	=	†	¤
0xd	ð	Ð	Ê	Ë	È	Ì	Í	Î	Ï	⋮	⋮	⋮	⋮	¡	ì	■
0xe	Ó	ß	Ô	Ò	Õ	µ	þ	þ	Ú	Û	Ü	Ý	Ý	Ý	–	’
0xf		±	=	¾	¶	§	÷	,	°	¨	.	¹	³	²	■	

## Code Page 874 Encoding – COUNTRY CP874

CP874, when used with CPF fonts acts transparently and characters simply pass through unmodified.

## How To Use For Single-Byte Encoding

The following fonts can be used for single-byte encoding. The table lists commonly encountered CPF fonts which may already be installed in your printer depending on the product configuration.

English/Latin/Cyrillic			CPCL Encoding Type
English/Latin (CP1252)	CPCL Built in Font	bitmap	CP1252 single-byte encoding can support CP850, CP874, FRANCE, GERMANY ITALY, LATIN9, NORWAY, SPAIN, SWEDEN, UK, USA  <b>NOTE:</b> SWIS721.CSF includes Latin, Cyrillic, Hebrew, Arabic glyphs. But it ONLY prints CP1252 glyphs due to single-byte encoding.
English/Latin (CP1252)	SWIS721.CSF	Scalable	
New Sans MT, Traditional Chinese	NSMTTC16.CPF	16x16 bitmap	
Cyrillic (CP1251)	DEJAVU12.CPF	12X12 bitmap	These fonts only support the Cyrillic code page. No encoding command is needed.
Cyrillic (CP1251)	DEJAVU14.CPF	14X14 bitmap	
Cyrillic (CP1251)	DEJAVU16.CPF	16X16 bitmap	
Cyrillic (CP1251)	DEJAVU20.CPF	20X20 bitmap	

Encodings available for built-in CPCL resident fonts:

Name	Encoding Type
CP850	USA with substitutions, single byte. Box and graphics chars not supported.
CP874	Font defines encoding for CPF.
FRANCE	USA with 7-bit substitutions, single byte.
GERMANY	USA with 7-bit substitutions, single byte.
ITALY	USA with 7-bit substitutions, single byte.
LATIN9	USA with substitutions, single byte.
NORWAY	USA with 7-bit substitutions, single byte.
SPAIN	USA with 7-bit substitutions, single byte.
SWEDEN	USA with 7-bit substitutions, single byte.
UK	USA with 7-bit substitutions, single byte.
USA	Font defines encoding, single byte.



**NOTE:** All lines in the following example label files are terminated with a carriage return and line feed.

### Example: Print the CSF font.

```
! 0 200 200 100 1
ST SWIS721.CSF 06 07 25 25 beleg sorgf
FORM
PRINT
```



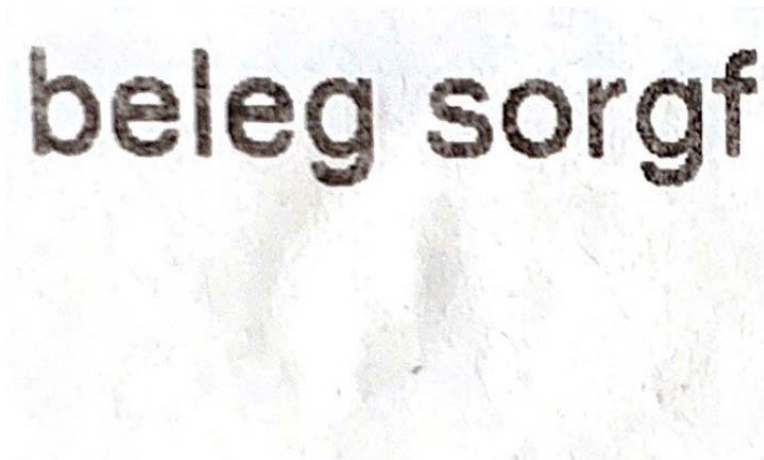
**Example: Print the CPF font.**

```
! 0 200 200 100 1
```

```
TEXT 4 0 25 25 beleg sorgf
```

```
FORM
```

```
PRINT
```



## Font Encodings – Multi-byte

In addition to the single byte encodings, there are also multi-byte encodings in CPCL.

### CHINA, JAPAN and KOREA Encodings

These encodings all specify data using GBK encodings.

When low-ASCII characters are encountered, they are referenced as a character from page 0 (size 0) of the font. When a high-ASCII character is encountered, the high bit is stripped off of the character and that value is saved as the first character of a multi-byte sequence. When a subsequent high-ASCII character is encountered, the high bit is also stripped off and saved as the second byte of the sequence. The first character is used as the size (character page) for the font and the second character is used as the character index into that page.

Some known issues with these encodings:

- If a low-ASCII character is encountered after a high-ASCII character, the low-ASCII character is printed and the high-ASCII first byte is still saved as the first character of a multi-byte sequence. When a subsequent high-ASCII character is received it will be treated as the second character of the sequence and printed.
- These encodings were designed around an older GBK encoding standard and do not support second byte characters < 0xA1 even though code page 936 as characters defined in these regions.

Example sequences:

- Only low-ASCII characters – The byte sequence 0x40, 0x41, 0x42 will print the characters 0x40, 0x41, and 0x42 from size (character page) 0x00.
- High-ASCII sequence – The byte sequence 0xCA, 0xFE will print the character 0x7E from size (character page) 0x4A.
- Interleaved low-ASCII character – The byte sequence 0xCA, 0x40, 0xFE will print the character 0x40 from size 0x00 followed by character 0x7E from size 0x4A.

### BIG5 Encoding

The BIG5 encoding is similar to the previous set of encodings in that it prints low-ASCII characters from font size (page) 0 and interprets any high-ASCII character as an entry into a two-byte sequence. The first character has the high bit stripped and is saved for use as the size (character page) to use for the character. The next character is used as the character index into the page. Unlike the previous encodings, the high bit is not stripped and there are no issues with interleaved low-ASCII characters as they are valid second characters in the sequence.

Example sequences:

- Only low-ASCII characters – The byte sequence 0x40, 0x41, 0x42 will print the characters 0x40, 0x41, and 0x42 from size (character page) 0x00.
- High-ASCII sequence – The byte sequence 0xCA, 0xFE will print the character 0xFE from size (character page) 0x4A.
- High-ASCII -> low-ASCII character – The byte sequence 0xCA, 0x40 will print the character 0x40 from size 0x4A.

## JAPAN-S (Shift-JIS) Encoding

The JAPAN-S country code is similar to BIG5 except that the high-ASCII characters 0xA0 through 0xDF, the half-width Kana characters, are not entry points into a multi-byte sequence but are printed from size (character page) 0.

## THAI Encoding

The THAI country code has 2 types of characters, single byte and double byte characters. The single byte characters are those that fall into these ranges inclusive, 0x10 – 0x20, 0x23 – 0x7E. These characters are referenced directly from size (character page) 0.

All other characters are the first byte of a 2 byte sequence. This encoding differs from the other encodings as they do not strip the high bit off the first or the second byte.

## VIETNAM Encoding

The VIETNAM encoding decodes a subset of UTF-8 characters, those used for the phonetic Vietnamese alphabet, and converts them into 2 byte sequences that are used to index into the CPF font files.

Single byte UTF-8 sequences (U+0000 through U+007F) are directly printed as that character from size (character page) 0.

Two-byte sequences (U+0080 through U+07FF) follow the binary form 110ABCDE 10FGHIJK where the letters can be either 0 or 1. These are then transformed into 2 bytes 00000ABC which is used as the size (character page) for the font and DEFGHIJK which is the character in that page.

Three-byte sequences (U+0800 through U+FFFF) follow the binary form 1110ABCD 10EFGHIJ.

10KLMNOP. These are transformed into 2 bytes 0BCDEFGH which is used as the size (character page) and IJKLMNOP which is used as the character index into that page.

## UTF-8

UTF-8 is the Unicode encoding that assigns each character code point to a sequence of one to four bytes.

## GB18030

GB18030 is a standard required by the People's Republic of China for operating systems of non-handheld computers.

The available multi-byte encodings are as follows:

Name	Encoding Type
BIG5	BIG5 Encoding.
CHINA	GBK Encoding.
JAPAN	GBK Encoding.
JAPAN-S	GBK Encoding.
KOREA	GBK Encoding.
THAI	Thai multi-byte encoding. Superseded by CP874.
VIETNAM	Font defines ECPF encoding.
UTF-8	
GB18030	

## Encoding Types for Common Fonts

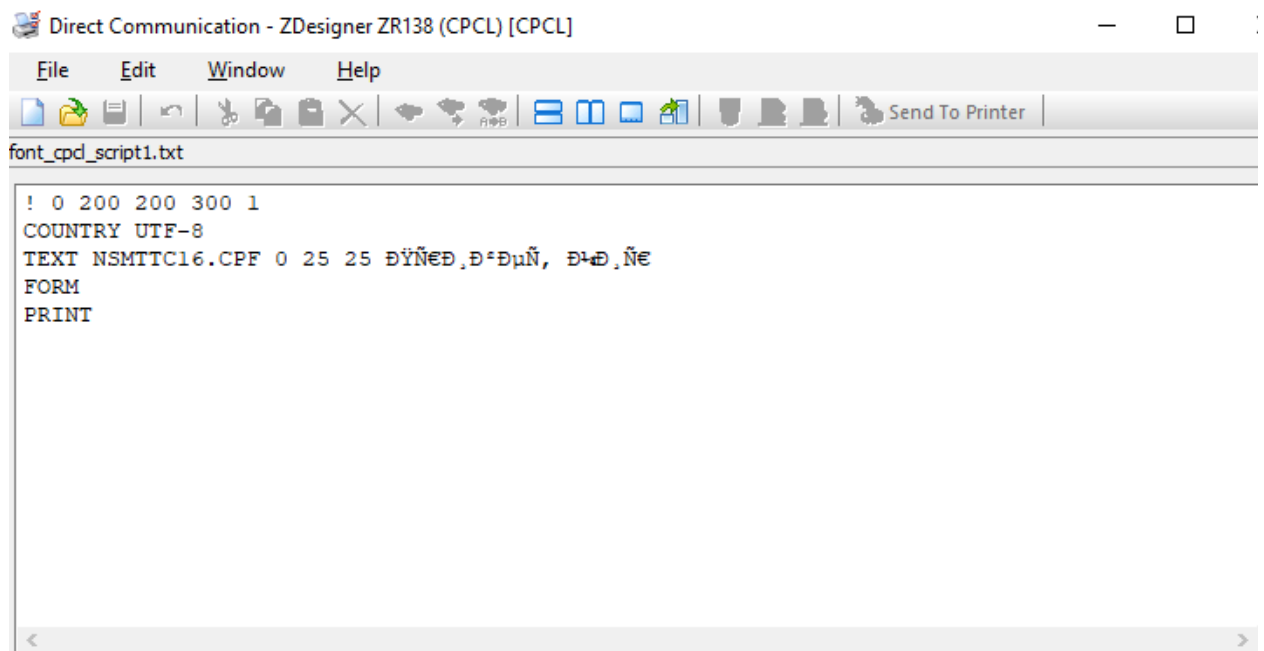
The following table lists the correct country to use for commonly encountered CPF fonts which may already be installed in your printer depending on the product configuration.

Simplified Chinese			CPCL Encoding Type
Sim Sun, Simplified Chinese	GBUNSG16.CPF	16x16 bitmap	UTF-8, GB18030, BIG5
M Sung, Simplified Chinese	GBUNSG24.CPF	24x24 bitmap	
Traditional Chinese			UTF-8, BIG5, GB18030
M Kai, Traditional Chinese	CTUNMK24.CPF	24x24 bitmap	
New Sans MT, Traditional Chinese	NSMTTC16.CPF	16x16 bitmap	
MingLiu, Traditional Chinese	MINGLIU.CPF	24x24 bitmap	
Vietnamese			
Utah, Vietnamese	MUTOS16.CPF	16x16 bitmap	UTF-8, VIETNAM
Others			
Code Page: 437, Latin 9, Latin 1, Latin 2, Cyrillic 2 and Cyrillic	FONTA/FONTB/FO NTC (Generate from NotoSans)	None-Monosp ace font	UTF-8

### Example: Print UTF-8 string via Zebra Setup Utility (ZSU)

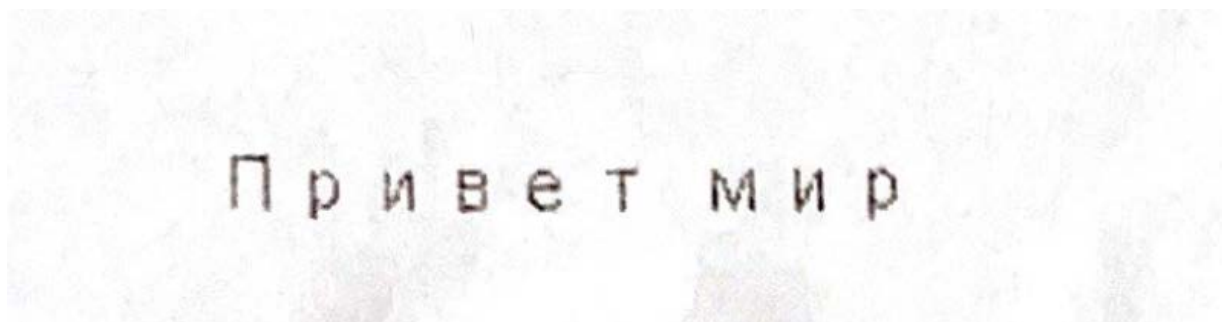
1. Prepare a test script and save it as file in UTF-8 format.  

```
! 0 200 200 300 1  
COUNTRY UTF-8  
TEXT NSMTTC16.CPF 0 25 25 Привет мир  
FORM  
PRINT
```
2. Open ZSU **Open Communication With Printer** and load the file you created in step 1.



3. Click **Send To Printer**.

Here is an example of the printout.



## Single-Byte Font Character List

Font: FONTA.CPF

Characters:

0020-00FF

Encoding: Single-Byte

Chars: 0020-00FF

```
0020 !"#$%&'()*+,-./
0030 0123456789:;<=>?
0040 @ABCDEFGHIJKLMNO
0050 PQRSTUVWXYZ[\]^_
0060 `abcdefghijklmno
0070 pqrstuvwxyz{|}~
0080
0090
00A0 ¡¢£¥¦§¨ª«¬®¯°±²³´µ¶·¸¹º»¼½¾¿
00B0  ¡¢£¤¥¦§¨©ª«¬­®¯°±²³´µ¶·¸¹º»¼½¾¿
00C0 ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏ
00D0 ÑÒÓÔÕÖ×ØÙÚÛÜÝÞß
00E0 àáâãäåæçèéêëìíîï
00F0 ðñòóôõö÷øùúûüýþÿ
```

Font: FONTB.CPF

Characters:

0020-00FF

Encoding: Single-Byte

Chars: 0020-00FF

```
0020 !"#$%&'()*+,-./
0030 0123456789:;<=>?
0040 @ABCDEFGHIJKLMNO
0050 PQRSTUVWXYZ[\]^_
0060 `abcdefghijklmno
0070 pqrstuvwxyz{|}~
0080
0090
00A0 ¡¢£¥¦§¨ª«¬®¯°±²³´µ¶·¸¹º»¼½¾¿
00B0  ¡¢£¤¥¦§¨©ª«¬­®¯°±²³´µ¶·¸¹º»¼½¾¿
00C0 ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏ
00D0 ÑÒÓÔÕÖ×ØÙÚÛÜÝÞß
00E0 àáâãäåæçèéêëìíîï
00F0 ðñòóôõö÷øùúûüýþÿ
```

## Font: FONTC.CPF

## Characters:

0020-00FF

Encoding: Single-Byte

## Chars: 0020-00FF

0020 !"#\$%&'()\*+,-./  
 0030 0123456789;:<=>?  
 0040 @ABCDEFGHIJKLMNO  
 0050 PQRSTUVWXYZ[\]^\_  
 0060 `abcdefghijklmnopqrstuvwxyz  
 0070 { } ~  
 0080  
 0090  
 00A0 ¡¢£¥¦§¨ª«¬®¯  
 00B0 °±²³´µ¶·¸¹º»¼½¾¿  
 00C0 ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎ  
 00D0 ÑÒÓÔÕÖ×ØÙÚÛÜÝÞß  
 00E0 àáâãäåæçèéêëìíî  
 00F0 ðñòóôõö÷øùúûüýþ

## Font: SWISS721.CSF

## Characters:

0020-00FF

Encoding: Single-Byte

## Chars: 0020-00FF

0020 !"#\$%&'()\*+,-./  
 0030 0123456789;:<=>?  
 0040 @ABCDEFGHIJKLMNO  
 0050 PQRSTUVWXYZ[\]^\_  
 0060 `abcdefghijklmnopqrstuvwxyz  
 0070 { } ~ ¢  
 0080 € Š š Ž Ÿ  
 0090 À æ Ē ž  
 00A0 ¡ ¢ £ ¤ ¥ ¦ § ¨ © ª « ® ¯  
 00B0 ° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿  
 00C0 À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î  
 00D0 Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß  
 00E0 à á â ã ä å æ ç è é ê ë ì í î  
 00F0 ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ



### Characters:

Encoding: Single-Byte

Chars: 0020-00FF

```

0020  !"#$%&'()*+,-./
0030  0123456789:;<=>?
0040  @ABCDEFGHIJKLMNO
0050  PQRSTUVWXYZ[\]^_
0060  `abcdefghijklmnopqrstuvwxyz
0070  {~
0080
0090
00A0      $ "
00B0      * ±      .
00C0      Ä Å      È É Ê
00D0      Ò Ó      ×
00E0      áâ      ëëê íí
00F0      óô      ÷ øú û

```

**Characters:**

Encoding: Single-Byte

Chars: 0020-00FF

```
0020 !"#%&'()*+,-./
0030 0123456789;:<=>?
0040 @ABCDEFGHIJKLMNO
0050 PQRSTUVWXYZ[\]^_
0060 `abcdefghijklmno
0070 pqrstuvwxyz{|}~
0080 ЁЂЃЄЅІЇЈЉЊЋЌЍЎ
0090 Џ'""—™љЎЋЌЍЎ
00A0 Ÿŷııı|şĖœ«»-ѐİ
00B0 °±İıµ¶·Ѽє»јѕѕї
00C0 АБВГДЕЖЗИЙКЛМНОП
00D0 РСТУФХЦЧШЩЪЫЬЭЮЯ
00E0 абвгдежзийклмноп
00F0 рстуфхцчшщъыьэюя
```

