Module: ZWIN-API.TXT
All commands for PPLZ in Dynamic Link Library for Windows

Version: 3.03 Revision date: 2005-04-07
Table of Contents:

No	Functions	Description
1	Z_Bar2d_Maxi()	Print Maxi code
2	Z_Bar2d_PDF417()	Print PDF-417
3	Z_ClosePrn()	Close printing task
4	Z_CreatePrn()	Create printing task
5	Z_Set_FlashMemory()	Select memory type to store form
6	Z_Set_Format_New()	Issue a new form name
7	Z_Del_Format()	Delete a form from DRAM or Flash
8	Z_Del_Graphic()	Delete a graphic from DRAM or Flash
9	Z_Draw_Box()	Draw a box
10	Z_Draw_Line()	Draw a line
11	Z_Get_DLL_Version()	Get library dll version
	Z_Get_DLL_VersionA()	Get library dll version value
12	Z_Get_Graphic_ColorBMP()	To transform BMP graphic into monochrome PCX graphic and select memory type to store the graphic
13	Z_Initial_Setting()	Initial load-in string or file
14	Z_Load_Graphic()	Recall graphic to print
15	Z_Open_ChineseFont()	Designate a file of dot-matrix Chinese font for function ”Z_Prn_Text_Chinese”.
16	Z_Print_Form()	Print a form (with label sets and copies settings)
17	Z_Print_Out()	Send command file of designated functions to printer
18	Z_Prn_Barcode()	Print a barcode
19	Z_Prn_Text()	Print fonts using a resident font type

20	Z_Prn_Text_Chinese()	Print Chinese fonts using a designated dot-matrix Chinese font file
21	Z_Prn_Text_TrueType() Z_Prn_Text_TrueType_W()	Print fonts using a true type font Print fonts using a true type font with font height and width adjustable
22	Z_Clear_Memory()	Clear Data stored in DRAM and Flash memory
23	Z_Set_Backfeed()	Set tear off position
24	Z_Set_Darkness()	Set printing darkness
25	Z_Set_DebugDialog()	Enable or disable debug environment
26	Z_Set_Label()	Set continuous label length (unit: dots)
27	Z_Set_Mode()	Set post-print action (tear off, peel off, cut)
28	Z_Set_Origin()	Set origin point of Y-Axis
29	Z_Set_Paper()	Printing with continuous media or non-continuous media
30	Z_Set_Prncomport	Set printer serial port
31	Z_Set_Prncomport_PC()	Set PC serial port
32	Z_Set_Reset()	Reset printer
33	Z_Set_Speed()	Set printing speed
34	Z_Set_TPH()	Set printing in direct thermal or thermal transfer mode
35	Z_Set_Unit()	Set basic measure unit (inch, mm, or dot)
36	Z_Set_Gap()	Set gap height (Only for media with gap > 0.5mm)
37	Z_Set_ProcessDlg()	Enable or disable printing task transmission process bar
38	Z_Bar2d_QRCode()	Print QR code
39	Z_Set_PrintWidth()	Set print width
40	Z_Print_OutQuality()	Set print quantity

Notice: Please copy all DLL files in Library3.03\PPLZ\Library\ to:

Win98: C:\windows\system

Win2000: C:\winnt\system32

WINXP: C:\windows\system32

20	Z_Prn_Text_Chinese()	Print Chinese fonts using a designated dot-matrix Chinese font file
21	Z_Prn_Text_TrueType() Z_Prn_Text_TrueType_W()	Print fonts using a true type font Print fonts using a true type font with font height and width adjustable
22	Z_Clear_Memory()	Clear Data stored in DRAM and Flash memory
23	Z_Set_Backfeed()	Set tear off position
24	Z_Set_Darkness()	Set printing darkness
25	Z_Set_DebugDialog()	Enable or disable debug environment
26	Z_Set_Label()	Set continuous label length (unit: dots)
27	Z_Set_Mode()	Set post-print action (tear off, peel off, cut)
28	Z_Set_Origin()	Set origin point of Y-Axis
29	Z_Set_Paper()	Printing with continuous media or non-continuous media
30	Z_Set_Prncomport	Set printer serial port
31	Z_Set_Prncomport_PC()	Set PC serial port
32	Z_Set_Reset()	Reset printer
33	Z_Set_Speed()	Set printing speed
34	Z_Set_TPH()	Set printing in direct thermal or thermal transfer mode
35	Z_Set_Unit()	Set basic measure unit (inch, mm, or dot)
36	Z_Set_Gap()	Set gap height (Only for media with gap > 0.5mm)
37	Z_Set_ProcessDlg()	Enable or disable printing task transmission process bar
38	Z_Bar2d_QRCode()	Print QR code
39	Z_Set_PrintWidth()	Set print width

Notice: Please copy all DLL files in Library3.03\PPLZ\Library\ to:

Win98: C:\windows\system

Win2000: C:\winnt\system32

WINXP: C:\windows\system32

- **1**
- Z_Bar2d_Maxi()

****

- **Function**: Print Maxi code
- **Syntax:**
  - int Z_Bar2d_Maxi(int x, int y, int nMode, int nSymbol, int nTotal, int nClass, int nCountry, char cZipCode1[6], char cZipCode2[4], LPCTSTR data, int increase)
- **Parameters:**

  Int x: x-axis

  Int y: y-axis

  Int nMode: Barcode mode

  2: Numeric Postal Code(US)

  3: Alphanumeric Postal Code(Non-US)

  int nSymbol: Symbol Number: 1 ~ 8

  int nTotal: Symbol Number Total: 1 ~ 8

  int nClass: three digit class of service

  int nCountry: three digit country code

  char cZipCode1[6]:

  nMode = 2 →five digit zip code

  nMode = 3 → six character zip code

  char cZipCode2:

  four digit zip code externsion.

  *Only used in nMode = 2.

  LPCTSTR data: Data strings, max. 84 characters

  int increase: increment, default value = 0
- **Return value:** 0 represents OK. Details please refer to ZW-Error.txt file.
- **Example:**

```
int  x , y, m, n, t, increase;
LPCTSTR data="This MaxiCode";
x=50; y=50; m=4; n=4; t=5, increase=0;
Z_Bar2d_Maxi(x,y,m,n,t,data,increase);
```

- **Comment**: This function Z_Bar2d_Maxi will print a Maxi Code.

- **2**
- Z_Bar2d_PDF417()

****

- **Function:** Print a PDF-417 code
- **Syntax:** int Z_Bar2d_PDF417(int x, int y, int o, int h, int s, int c, int r, int t, int narrow, LPCTSTR data, int increase)
- **Parameters:**
  - **int x:** x-axis
  - **int y:** y-axis
  - **int o:** printing orientation, 1=0°, 2=90°, 3=180°, 4=270°
  - **int h:** barcode height for individual rows
    h * module = height of individual rows
  - **int s:** error correction level (0~8)
  - **int c:** column count
  - **int r:** row count
  - **int t:** truncation flag, 0=normal; 1=truncated
  - **int narrow:** module height
  - **LPCTSTR data:** data string, max. 84 characters
  - **int increase:** increment, default=0
- **Return value:** 0→OK. Details please refer to ZW-Error.txt.
- **Example:**

```
int   x, y, o, h, s, c, r, t, narrow, increase;
LPCTSTR data=”PDF-417”;
x=50; y=50; o=0; h=5; s=5; c=2; r=83; t=0; increase=0; narrow=2;
Z_Bar2d_PDF417(x, y, o, h, s, c, r, t, narrow, data, increase);
```

- **Comment:** Above function will print a PDF-417 code.

- **3**
- Z_ClosePrn()

****

- **Function:** Close printing task
- **Syntax:** void Z_CkosePrn(void);
- **Example:**

  Z_ClosePrn();
- **Comment:**

  The function Z_ClosePrn will release the memory being used, close the output port, or close the printing file you opened. This function must be located at the end of all functions.

- **4**
- Z_CreatePrn()

****

- **Function:** Create a printing task
- **Syntax:** int Z_CreatePrn(int selection, LPCTSTR filename);
- **Parameters:**

Int selection: select the output port or print file

0→ print to file
1→ lpt1:
2→ lpt2:
3→ lpt3:
4→ com1:
5→ com2:
6→ com3:
10→ net directory

LPCTSTR filename:

If int selection="0", data will be outputted as file, file name will be the what you defined. Path or Null value could be included.

If int selection="10", file name will be the output path / directory.

Return value: 0→OK. (Refer to ZW-Error.txt.)

Example:

Z_CreatPrn(1,NULL);
Comment: print through parallel port
Z_CreatPrn(0,"c:\TEMP\output.prn");
Comment: print to file, file name →output.prn
Z_CreatPrn(10,"\\paul\LabelDr.200");

- **Comment:**

print through network shared printer, the shared printer name is LabelDr.200

- **Notice:** This function must be positioned before all other functions.

- **5**
- Z_Set_FlashMemory()

****

- **Function:** Set memory type to store form
- **Syntax:** int Z_Set_FlashMemory(int Status);
- **Parameters:**

  Int Status: designate memory type for form storage.

  0→ RAM, 1→ Flash Memory
- **Return value:** 0→OK (Refer to ZW-Error.txt)
- **Example:**

  Z_Set_FlashMemory(0);

  Store form to DRAM

  Z_Set_FlashMemory(1);

  Store form to Flash
- **Comment:**

  Use this function to designate the memory type to store form. The function must precede the form storing function.

- **6**
- Z_Set_Format_New()

****

- **Function:** Set form name
- **Syntax:** void Z_Set_Format_New(LPCTSTR FormName);
- **Parameters:**

  LPCTSTR FormName: Form Name
- **Example:**

  Z_Set_Format_New(“demo”);
- **Comment:**

  This function sets the form name. The form could be recalled to print after setting.

- **7**
- Z_Del_Format()

****

- **Function:** Delete a form
- **Syntax:** int Z_Del_Format(int memory, char FormName[])
- **Parameters:**

  Int memory: designate memory type to store a form

  0→DRAM; 1→Flash
- **Example:**

  Z_Del_Format(0,"demo");
- **Comment:**

  The above function will delete a form named "demo".

- **8**
- Z_Del_Graphic()

****

- **Function:** Delete a graphic from DRAM or Flash
- **Syntax:** int Z_Del_Graphic(int memory, char GraphicName[])
- **Parameters:**

  Int memory: designate memory type to store a graphic

  0→DRAM; 1→Flash

  char GraphicName[ ]: graphic name (up to 10 alphanumeric characters)
- **Example:**

  Z_Del_Graphic(0,"girl");
- **Comment:**

  The above sample will delete a graphic named "girl" from DRAM.

- **9**
- Z_Draw_Box()

****

- **Function:** Draw a box
- **Syntax:** int Z_Draw_Box(int x, int y, int width, int height, int thickness)
- **Parameters:**
  - **int x:** x-axis
  - **int y:** y-axis
  - **int width:** width of a box (0<width<9999)
  - **int height:** height of a box (0<height<9999)
  - **int thickness:** border thickness (1 dot = 0.125mm)
- **Example:**

```
Int x, y , width, height, thickness;
x=50; y=100; width=250; height=100; thickness=5;
Z_Draw_Box(x, y, width, height, thickness);
```

- **Comment:** Above function will draw a box.

- **10**
- Z_Draw_Line()

****

- **Function:** Draw a line
- **Syntax:** int Z_Draw_Line(int x, int y, int width, int height)
- **Parameters:**
  - **int x:** x-axis
  - **int y:** y-axis
  - **int width:** line width (0<width<9999)
  - **int height:** line height (0<height<9999) (1 dot =0.125mm)
- **Example:**

```
Int x, y, ,width, height;
X=50; y=100; width=250; height=3;
Z_Draw_Box(x, y, width, height);
```

- **Comment:** Above function will draw a line

- **11**
- Z_Get_DLL_Version()
- Z_Get_DLL_VersionA()

****

- **Function:** Retrieve or show library DLL version
- **Syntax:**
  - char Z_Get_DLL_Version(int nShowMessage);
  - int Z_Get_DLL_VersionA(int nShowMessage);
- **Parameters:**

  Int nShowMessage: message window prompts function

  0→disable; 1→enable
- **Return value:** return the string of present version, if failed then return NULL.

  Z_Get_DLL_VersionA()→return version value
- **Example:**

  Z_Get_DLL_Version(1);
- **Comment:**

  Above function will prompt a message window and show present library DLL version.

- **12**
- Z_Get_Graphic_ColorBMP()

****

- **Function:** Transfer colored BMP graphics to monochrome BMP graphics, and store the graphic in DRAM or Flash Memory.
- **Syntax:** int Z_Get_Graphic_ColorBMP(int x, int y, int memory, LPCTSTR filename);
- **Parameters:**

  int x: x-axis

  int y: y-axis

  int memory: designate memory type to store graphics.

  0→DRAM; 1→Flash Memory

  LPCTSTR filename: Graphic name (or including path)

  The parameter format: i.e. XXXXXXXX.XXX or X:\XXX\XXX.BMP
- **Return value:** 0→OK (Refer to ZW_Error.txt)
- **Example:**

  Z_Get_Graphic_ColorBMP(30,20,"girl.bmp");
- **Comment:**

  Via this function you can load in bmp graphics with any color type, it will be transferred to monochrome BMP graphics. The transferred graphic quality depends on the driver (Label Dr.200 or Label Dr.300) you installed. One of the two drivers must be installed and better set as default printer. If Label Dr.200 or Dr.300 is not set as default printer, the program will auto-search and detect the two drivers in sequence for use.

- **13**
- Z_Initial_Setting()

****

- **Function:** Initial load-in strings or file
- **Syntax:** int Z_Initia_Setting(int Type, LPCTSTR Source);
- **Parameters:**

  int Type: select the input type

  0→ input strings

  1→ input file

  LPCTSTR Source: data source, could be strings or file name (or including path)
- **Return value:** 0→OK. (Refer to ZW-Error.txt)
- **Example:**

```
LPCTST aa="^FO120,30\r\n\"; Z_Initial_Setting(0,aa);
Z_Initial_Setting(1,"initfile.txt");
```

- **Comment:**

  This function will send a command strings or file to printer. The function must be positioned after function Z_Creat_Prn() but precede other functions to be used as customized settings.

- **14**
- Z_Load_Graphic()

****

- **Function:** Recall and print a graphic
- **Syntax:** int Z_Load_Graphic(int x, int y, char GraName[11], int hori, int vert);
- **Parameters:**

  int x: x-axis

  int y: y-axis

  char GraName: name of stored image(exclude file name extension), up to 10 alphanumeric characters.

  int hori: magnification on the x-axis (1<= hori <=10, default = 1)

  int vert: magnification on the y-axis (1<= vert <=10, default = 1)
- **Return value:** 0→OK (Refer to ZW-Error.txt)
- **Example:**

```
int x, y, hori, vert;
char GraName[11]=”phone”;
x=50, y=30, hori=1, vert=1;
Z_Load_Graphic(x, y, GraName, hori, vert);
```

- **Comment:**

  Above function will print a graphic, but Z_Get_Graphic function should be preceded executed to recall the graphic stored in DRAM or Flash Meomory.

- **15**
- Z_Open_ChineseFont()

****

- **Function:** Designate a file of dot-matrix Chinese font
- **Syntax:** int Z_Open_ChineseFont(char*path);
- **Parameters:**

  Char*path: Path where the dot-matrix Chinese font file locates
- **Return value:** 0→OK (Refer to ZW-Error.txt)
- **Example:** Z_Open_ChineseFont("C:\ET3");
- **Comment:**

  Function will open a dot-matrix Chinese font file (16*15 and 24*24) for usage of function Z_Prn_Text_Chinese.

- **16**
- Z_Print_Form()

****

- **Function:** Print a form (with label sets and counter settings)
- **Syntax:** int Z_Print_Form(int labelset, int copies, int mem, char form_out[ ]);
- **Parameters:**
  - **int labset:** total quantity of labels to print (range: 1~32767)
  - **int copies:** replicates of each serial number (1~32767)
  - **int mem:** memory type to retrieve a form
  - **char form_out[11]:** form name that is previously loaded or stored in printer
- **Return value:** 0→OK (Refer to ZW-Error.txt)
- **Example:**

```
int labset, copies, mem
char form_out[11]=”demo”;
labset=3; copies=2, mem=0;
Z_Print_Form(labset, copies, mem, form_out);
```

- **Comment:**

  The function will perform the print task and send a form to printer. Before use of this function, Z_Set_Format_New() must be prior executed, and Z_Print_Form() should be positioned before Z_ClosePrn, but after all other functions.

  When use this function, function Z_Print_Out() will not be required.

- **17**
- Z_Print_Out()

****

- **Function:** perform the printing task
- **Syntax:** int Z_Print_Out(int sets, int copies);
- **Parameter:**
  int sets: total quantity of labels to print
  int copies: replicates of each serial number
- **Return value:** 0→OK (Refer to ZW-Error.txt)
- **Example:** Z_Print_Out(1,1);
- **Comment:**
  This function will perform the print task, all designated functions and parameters will be transferred to PPLZ printer commands and sent to printer. Please position this function before function Z_ClosePrn().

- **18**
- Z_Prn_Barcode()

****

- **Function:** print a barcode (with counter function)
- **Syntax:** Z_Prn_Barcode(int x, int y, int ori, int narrow, int width, int height, char type, int increase, LPCTSTR data, char p1, char p2, char p3, char p4, char p5)
- **Parameters:**

int x: x-axis
int y: y-axis (203 dpi→1dot = 0.125mm)
int ori: orientation
1 → normal
2 → rotated 90 degree (clockwise)
3 → inverted 180 degree
4 → read from bottom up, 270 degree
int narrow: narrow bar width
int width: wide bar width
int height: bar code heigth

- **char type:** bar code type (refer to below table→Type)

int increase: increment
LPCTSTR data: barcode data
Char p1, p2, p3, p4, p5: (refer to below table→Details)

Type	Bar code type	Details (Char p1~p5)
1	Code 11 Bar Code	p1: print check digit (Y: 1 digit, N: 2 digits) p2: print interpretation line (Y or N) p3: print interpretation line above code (Y or N)
2	Interleaved 2 of 5 Bar Code	p1: print interpretation line (Y or N) p2: print interpretation line above code (Y or N) p3: print check sum (Y or N)
3	Code 39 Bar Code	p1: print check digit (Y or N) p2: print interpretation line (Y or N) p3: print interpretation line above code (Y or N)
8	EAN-8 Bar Code	p1: print interpretation line (Y or N) p2: print interpretation line above code (Y or N)
9	UPC-E Bar Code	p1: print interpretation line (Y or N) p2: print interpretation line above code (Y or N) p3: print check digit (Y or N)

Type	Bar code type	Char p1, p2, p3, p4, p5 details
A	Code 93 Bar Code	p1: print interpretation line (Y or N) p2: print interpretation line above code (Y or N) p3: print check digit (Y or N)
C	Code 128 Bar Code	p1: print interpretation line (Y or N) p2: print interpretation line above code (Y or N) p3: UCC check digit (Y or N) p4: mode (N: Code 128, U: Code 128 UCC)
E	EAN-13 Bar Code	p1: print interpretation line (Y or N) p2: print interpretation line above code (Y or N)
I	Industrial 2 of 5 Bar Code	p1: print interpretation line (Y or N) p2: print interpretation line above code (Y or N)
J	Standard 2 of 5 Bar Code	p1: print interpretation line (Y or N) p2: print interpretation line above code (Y or N)
K	ANSI Codabar Bar Code	p1: print check digit (Y or N) p2: print interpretation line (Y or N) p3: print interpretation line above code (Y or N) p4: designate start character (accepted value: A, B, C, D) p5: designate stop character (accepted value: A, B, C, D
L	LOGMARS Bar Code	p1: print interpretation line above code (Y or N)
M	MSI Bar Code	p1: check digit selection A: no check digits B: 1 Mod 10 C: 2 Mod 10 D: 1 Mod 10 and 1 Mod 11 p2: print interpretation line (Y or N) p3: print interpretation line above code (Y or N) p4: print check digit (Y or N)
S	UPC/EAN Extensions	p1: print interpretation line (Y or N) p2: print interpretation line above code (Y or N)
U	UPC-A Bar Code	p1: print interpretation line (Y or N) p2: print interpretation line above code (Y or N) p3: print check digit (Y or N)
Z	POSTNET Bar Code	p1: print interpretation line (Y or N) p2: print interpretation line above code (Y or N)

Notice: if a parameter is not used, any value could be inputted.

- **19**
- Z_Prn_Text()

****

- **Function:** print text with built-in fonts (with counter function)
- **Syntax:** int Z_Prn_Text(int x, int y, int ori, char font, int height, int width, int increase, LPCTSTR data);
- **Parameters:**

  int x: x-axis

  int y: y-axis (for 203 dpi, 1 dot = 0.125mm)

  int ori: orientation.

  1 → normal

  2 → rotated 90 degree (clockwise)

  3 → inverted 180 degree

  4 → read from bottom up, 270 degree

  char font: Built-in font type selection (A~H: bitmapped fonts; 0, P~V: scalable fonts / smooth vector fonts)

  # **Bitmapped Font Size:**

Font ID	A	B	C, D	E	F	G	H
H * W	9 * 5	11 * 7	18 * 10	28 * 15	26 * 13	60 * 40	21 * 13

  int height: magnification of height (##)

  int width: magnification of width (##)

  ## ***Bitmapped fonts (A~H):*** Multiple of height / width from 2 to 10 times the standard height / width in increments of 1.

  ## ***Scalable fonts (0, P~V):*** 10~1500 dots.

  int increase: increment of counter

  LPTCSTR data: text data

- **Return value:** 0→OK (Refer to ZW-Error.txt)
- **Example:**

```
Z_Prn_Text(50, 110, 0, 'A', 90, 90, 0, "A123456");
or
Z_Prn_Text(50, 110, 0, 'P', 27, 360, 0, "A123456");
```

- **20**
- Z_Prn_Text_Chinese()

****

- **Function:** print a text with dot-matrix Chinese font (16*15 or 24*24)
- **Syntax:** int Z_Prn_Text_Chinese(int x, int y, int fonttype, LPCTSTR id_name, LPCTSTR data, int mem);
- **Parameters:**
  - **int x:** x-axis (dot)
  - **int y:** y-axis (dot)
  - **int fonttype:** font type of dot-matrix Chinese font (0→16*15, 1→24*24)
  - **LPCTSTR id-name:** assign a font ID to the font type to store in printer, then use function Z_Load_Graphic() to recall the font.
  - **LPCTSTR data:** text data
  - **Int mem:** memory device to store the font (0→DRAM, 1→ Flash memory)
- **Return value:** 0→OK (Refer to ZW-Error.txt)
- **Example:**

```
Z_Open_ChineseFont("C:\\ET3");
Z_Prn_Text_Chinese(100, 100, 0, "A01", "中文文字=Chinese characters", 0);
```

- **Comment:**

  Before use of this function, Argox windows driver Label Dr.200 or Label Dr.300 must be installed and better be set as default printer.

  Printing darkness is set in driver.

  If Label Dr.200 or Dr.300 is not set as default printer, the program will auto-search and detect the two drivers in sequence for use.

- **21**
- Z_Prn_Text_TrueType()
- Z_Prn_Text_TrueType_W()

****

- **Function:** Create a printing task
- **Syntax:**

  int Z_Prn_Text_TrueType(int x, int y, int FSize, LPCTSTR Ftype, int Fspin, int FWeigth, int FItalic, int FUnline, int FStrikeOut, LPCTSTR id-name, LPCTSTR data, int mem);

  int Z_Prn_Text_TrueType_W(int x, int y, int FHeight, int FWidth, LPCTSTR Ftype, int Fspin, int FWeigth, int FItalic, int FUnline, int FStrikeOut, LPCTSTR id-name, LPCTSTR data, int mem);
- **Parameters:**

  int x: x-axis

  int y: y-axis (1 dot = 0.125mm)

  int FSize: True type font size (unit: dot). FSize = (dpi * point) / 72.

  int FHeight: Font height (unit: dot). FHeight = (dpi * point) / 72

  int FWidth: Font width (unit: dot). FWidth = (dpi * point) / 72

  LPCTSTR FType: True type font typeface

  int FSpin: orientation
  1→normal, 2→rotated 90 degree [clockwise], 3→inverted 180 degree, 4→read from bottom up, 270 degree).

  int FWeight: True type font boldness
  0 and NULL and 400 →regular
  100→extra fine, 200→very fine
  300→fint, 500→normal
  600→half bold, 700→bold
  800→extra bold, 900→boldface

  int FItalic: True type font in italic style.
  0→False, 1→True.

  int FUnline: True type font with underline.
  0→Flase, 1→True.

  int FStrikeOut: True type font with delete-line (strike-line).
  0→False, 1→True.

  LPCTSTR idname: assign a font ID to the true type font to store in printer, then use function Z_Load_Graphic() to recall the font.

  LPCTSTR data: text data

  int mem: memory type to store the font (0→DRAM, 1→Flash)

- **Example:**

  Z_Prn_Text_TrueType(30, 35, 40, ”Arial”, 4, 400, 0, 0, 0, “AA”, “True Type Font Test”, 0);

  Z_Prn_Text_Truetype_W(30, 35, 40, 30, ”Times New Roman”, 4, 400, 0, 0, 0, ”AB”, “True Type Font Test”, 0);

- **Comment:**

  Argox windows driver Label Dr.200 or Label Dr.300 must be installed and better be set as default printer.

  Printing darkness is set in driver.

  If Label Dr.200 or Dr.300 is not set as default printer, the program will auto-search and detect the two drivers in sequence for use.

- **22**
- Z_Clear_Memory()

****

- **Function:** Clear data stored in printer DRAM or Flash Memory.
- **Syntax:** void Z_Clear_Memory(void);
- **Example:** Z_Clear_Memory();
- **Comment:**

  This function will erase all graphics and soft fonts stored in printer DRAM or Flash Memory due to graphics and soft fonts will accumulate in memory that will cause printer memory overflow.

  The function should be sent to printer before function Print_Out().

  While printer memory is full, the previous loaded graphics and soft fonts will be erased from DRAM or Flash Memory.

  To avoid improper deletion, the function Z_Clear_Memory should be sent to printer before function Z_Print_Out().

- **23**
- Z_Set_Backfeed()

****

- **Function:** Set tear off position
- **Syntax:** void Z_Set_Backfeed(int nPixel);
- **Parameter:**

  int pixel: set the tear off distance, range: 0~999, unti: dot.
- **Example:** Z_Set_Backfeed(204);
- **Comment:**

  Above function example will enable back feed with a distance of 204 dots.

- **24**
- Z_Set_Darkness()

****

- **Function:** Set printing darkness
- **Syntax:** int Z_Set_Darkness(int darkness);
- **Parameter:**

  int darkness: set printing darkness, range:0~30. Factory default value is 0.
- **Return value:** 0→OK (Refer to ZW-Error.txt)
- **Example:** Z_Set_Darkness(12);
- **Comment:**

  This function is to control the printing darkness, to obtain better printing quality, other factors such as label material, ribbon types, and the image pattern itself should be considered as well.

➢ Z_Set_DebugDialog()

****

- **Function:** Set debug environment and warning dialog
- **Syntax:** int Z_Set_DebugDialog(int nEnable);
- **Parameter:**

  int nEnable: 1→ enable debug environment, 0→ disable debug environment.
- **Return value:** 0→OK (Refer to ZW_Error.txt)
- **Example:** Z_Set_DebugDialog(1);
- **Comment:**

  This function will set a debug environment for a program, except returning relevant error code, it will also show a warning dialog.

- **26**
- Z_Set_Label()

****

- **Function:** Set continuous label length
- **Syntax:** int Z_Set_Label(int Length);
- **Parameter:**
  int length: continuous label length (1 dot = 0.125mm)
- **Return value:** 0→OK. (Refer to ZW-Error.txt)
- **Example:** Z_Set_Label(300);
- **Comment:** Above example will set continuous label length to 300 dots.

- **27**
- Z_Set_Mode()

****

- **Function:** Set print mode (tear off mode, peel off mode, cutting mode)
- **Syntax:** int Z_Set_Mode(char mode);
- **Parameter:**
  char mode: T: tear off, P: peel off, C: cutter enable
- **Return value:** 0→ OK (Refer to ZW-Error.txt)
- **Example:** Z_Set_Mode('C');
- **Comment:** This is an important function in a program.

- Z_Set_Origin()

****

- **Function:** Redefine the origin point of y-axis.
- **Syntax:** int Z_Set_Origin(int y);
- **Parameter:**
  int y: new origin point of y-axis
- **Return value:** 0→OK. (Refer to ZW-Error.txt)
- **Example:** Z_Set_Origin(10);

- **29**
- Z_Set_Paper()

****

- **Function:** Set label type as continuous media or non-continuous media
- **Syntax:** int Z_Set_Paper(char p);
- **Parameter:**

  Char p: N: continuous media. Y: non-continuous media
- **Return value:** 0→OK (Refer to ZW-Error.txt)
- **Example:** Z_Set_Paper('Y');

- **30**
- Z_Set_Prncomport()

****

- **Function:** Set printer serial communications
- **Syntax:** int Z_Set_Prncomport(int baud, int parity, int data, int stop);
- **Parameters:**
  - **int baud:** accepted values: 2400; 4800; 9600; 19200; 38400
  - **int parity:** parity. 0: none parity; 1: even parity; 2: odd parity
  - **int data:** data bit (7 or 8)
  - **int stop:** stop bit (1 or 2)
- **Return value:** 0→OK. (Refer to ZW-Error.txt)
- **Example:**

```
Int baud, data, stop;
Char parity;
Parity = 'N'; baud=93; data=8; stop=1;
Z_Set_Prncomport(baud, parity, data, stop);
```

- **Comment:**

  Printer and PC must have the same serial communication settings.

- **31**
- Z_Set_Prncomport_PC()

****

- **Function:** Set PC serial communications
- **Syntax:** int Z_Set_Prncomport_PC(int nBaudRate, int nByteSize, int nParity, int nStopBits, int nDsr, int nCts, int nXonXoff);
- **Parameters:**

int nBaudRate: Baud rate

1→	110;	9→	19200
2→	300;	10→	38400
3→	600;	11→	56000
4→	1200;	12→	57600
5→	2400;	13→	115200
6→	4800;	14→	128000
7→	9600;	15→	256000
8→	14400;	0→	9600

int nByteSize: Data bit
0→ 7-bit data
7→ 7-bit data
8→ 8-bit data

int nParity: Parity
0→ none parity
1→ even parity
2→ odd parity

int nStopBits: Stop bit
0→ 1 stop bit
1→ 1 stop bit
2→ 2 stop bit

int nDsr: set hardware flow control
1→ DTR control handshake
0→ DTR control enable

int nCts: set hardware flow control
1→ RTS control handshake
0→ RTS control enable

int nXonXoff: Set software flow control
0→ enable; 1→ disable

- **Return value:** 0→OK. (Refer to ZW-Error.txt)
- **Example:** Z_Set_Prncomport_PC(0,0,0,0,1,1,1);
- **Comment:**

  Printer and PC must have the same serial communication settings. This function must be positioned before function Z_Print_Out().

- **32**
- Z_Set_Reset()

****

- **Function:** reset printer to factory default
- **Syntax:** void Z_Set_Reset();
- **Example:** Z_Set_Reset();

- **33**
- Z_Set_Speed()

****

- **Function:** Set printing speed
- **Syntax:** int Z_Set_Speed(int s);
- **Parameter:**
  int s: 1~6 (IPS)
- **Return value:** 0→OK (Refer to ZW-Error.txt)
- **Example:** Z_Set_Speed(3);
- **Comment:** Above function will set printing speed at 3 IPS.

- **34**
- Z_Set_TPH()

****

- **Function:** Set media type (thermal transfer media or direct thermal media)
- **Syntax:** int Z_Set_TPH(char m);
- **Parameter:**
  char m: T: thermal transfer media; D: direct thermal media
- **Return value:** 0→OK (Refer to ZW-Error.txt)
- **Example:** Z_Set_TPH('T');
- **Comment:**
  # Above function will set media type as thermal transfer mode.
  # This function is the same as setting printing in thermal transfer mode or direct thermal mode.
  # This is an important function in a program.

- **35**
- Z_Set_Unit()

****

- **Function:** Set units of measurement (inch, mm, dots)
- **Syntax:** int Z_Set_Unit(char m);
- **Parameters:**

  char m: D: dots; I: inches; M: millimeters
- **Return value:** 0→OK (Refer to ZW-Error.txt)
- **Example:** Z_Set_Unit('D');

- **36**
- Z_Set_Gap()

****

- **Function:** set label gap length
- **Syntax:** int Z_Set_Gap(int nPattern, int gapLen);
- **Parameters:**
  int nPattern: 0~65535 (mm)
  int gapLen: 5~65535 (mm)
- **Return value:** 0->OK (Refer to ZW-Error.txt)
- **Example:** Z_Set_Gap(30,20);
- **Comment:**
  This function is used only when label gap length is over 5mm, and the function should be positioned before function Z_Set_Label().

- **37**
- Z_Set_ProcessDlg()

****

- **Function:** Enable or disable printing task transmission process bar
- **Syntax:** int Z_Set_ProcessDlg(int nShow);
- **Parameters:**
  int nShow: 0→disable; 1→ enable.
- **Return value:** 0→OK (Refer to ZW-Error.txt)
- **Example:** Z_Set_ProcessDlg(1);

- **38**
- Z_Bar2d_QRCode()

****

- **Function:** print QR code
- **Syntax:** int Z_Bar2d_QRCode(int x, int y, int nModel, int nMagni, int nErr_Cor, int nInput, LPCTSTR data, int increase);
- **Parameters:**
  - **int x:** x-axis
  - **int y:** y-axis
  - **int nModel:** accepted values: 1→original; 2→enhanced
  - **int nMagni:** magnification factors
    - 1→ on 150 dpi printers
    - 2→ on 200 dpi printers
    - 3→ on 300 dpi printers
    - 4→ through 10
  - **int nErr_Cor:** error correction level
    - 1→ ultra-high reliability level
    - 2→ high reliability level
    - 3→ standard (default)
    - 4→ high density level
  - **int nInput:** 1→ automatic; 2→ manual
  - **LPCTSTR data:** barcode data
  - **Int increase:** increment
- **Return value:** 0→OK (Refer to ZW-Error.txt)
- **Example:** int Z_Bar2d_QRCode(20,20,2,"QR Code",0);

- **39**
- Z_Set_PrintWidth()

****

- **Function:** Set print width
- **Syntax:** int Z_Set_PrintWidth(int nDotWidth);
- **Parameter:**

  int nDotWidth: print width in dots
- **Return value:** 0→OK (Refer to ZW-Error.txt)
- **Example:** int Z_Set_PrintWidth(406);
- **Comment:**

  Above function will set print width to 406 dots (2 inches).

- **40**
- Z_Print_OutQuality()

****

- **Function:** Set print quantity
- **Syntax:** int Z_Print_OutQuality(int nTotal, int copies, int sets, int nPause);
- **Parameter:**
  - **int nTotal:** total quantity of labels to print
  - **int sets:** pause and cut value (labels between pauses)
  - **int copies:** replicates of each serial number
  - **int nPause:** override pause count. 1 --> Yes, 0 --> No
- **Return value:** 0→OK (Refer to ZW-Error.txt)

- **Example:** Z_Print_OutQuality(1,1,1,1);
- **Comment:**

  The function Z_Print_OutQuality will send the printing task to printer. The function must be positioned after all other functions but precedes function Z_ClosePrn().