1 Introduction

Code samples introduced here will be in VC++, tested in the eVC++ from Microsoft.

2 File contents

☞ WoosimPrinter.DLL (Ver. 2.0.1.2)

3 Import function

☞ int InitPrinter(TCHAR* sPortName, int iBaudRate, BOOL bProtocol)

☞ int InitWlanPrinter(TCHAR *sIP_ADDR, int iPortNum, int iTimeout, BOOL bProtcol)

☞ BOOL Close_Printer()

☞ int ControlCommand(BYTE *Cmd, int iLength)

☞ int SaveSpool(TCHAR*sData,int iFontValue, BOOL bEmphasis)

☞ int PrintSpool(BOOL bDelete_Spool)

☞ void ClearSpool()

☞ void CardRead(BOOL bTrack2)

☞ void CardCancel()

☞ int BmpSaveSpool(TCHAR* BmpFilePath, int iXPosition, int iYPosition)

☞ int BarcodeSaveSpool(BYTE ucBarcodeType, BYTE ucBarWidth,BYTE ucBarHeight,
BOOL HRI_On, TCHAR* BarcodeData, int iPrtWidth,int iPrtHeight,int iXPos)

4 Message and return value

n **Message**

☞ MSG_RECEIVE_CHECK _T("WOOSIM_PRT_OK")

UWM_RECEIVE_CHECK = ::RegisterWindowMessage(MSG_RECEIVE_CHECK)

n **Message Parameter(WPARAM Parameter Value)**

☞ NACK 0x15

☞ DATA 0x44

☞ EOT 0x04

☞ ETX 0x03

☞ MSR_FAIL 0x4D

n **Return value**

< BLUETOOTH >

☞ SUCCESS	1
☞ ALREADY_OPENED	-1
☞ UNABLE_TO_OPEN_THE_PORT	-2
☞ UNABLE_TO_CONFIGURE_THE_SERIAL_PORT	-3
☞ UNABLE_TO_SET_THE_TIMEOUT_PARAMETERS	-4
☞ NO_RESPONSE_FROM_PRINTER	-5

< WLAN >

☞ SUCCESS	1
☞ SOCKET_ERROR	-1
☞ CONNECT_FAIL	-2
☞ ALREADY_CONNECTED	-3
☞ NOT_OPEN_THE_PORT	-11
☞ BUFFER_OVERFLOW	-12
☞ OPERATOR_ERROR	-13

n **Barcode Type**

☞ UPCA	65
☞ UPCE	66
☞ EAN13	67
☞ EAN8	68
☞ CODE39	69
☞ ITF	70
☞ CODABAR	71
☞ CODE93	72
☞ CODE128	73

n **Print Width**

☞ _2INCH	384
☞ _3INCH	576
☞ _4INCH	832

5 Function Reference

n **int InitPrinter(TCHAR *sPortName, int iBaudRate, BOOL bProtocol)**

: This function opens the printer communications resource.

Parameter

sPortName

Specifies the name of a communications resource to open.

you must include a colon after the name.

For example, specify "COM1: " to open that port.

iBaudRate

Specifies the baud rate at which the communication device operates. It is an actual baud rate value

bProtocol

Specifies if protocol mode is enabled. If this member is TRUE, protocol mode is performed and the printer status are reported.

For example, Protocol IrDA, Protocol Bluetooth, Protocol RS232

- **Return value**

If the port is opened successfully and there is response from printer,

▶ SUCCESS ,

If called more than twice,

▶ ALREADY_OPENED

If the communication port is already opened from another program,

▶ UNABLE_TO_OPEN_THE_PORT

If the time out setting is failed,

▶ UNABLE_TO_SET_THE_TIMEOUT_PARAMETERS,

If there is no response from printer,

▶ NO_RESPONSE_FROM_PRINTER;

n **int InitWlanPrinter(TCHAR *sIP_ADDR, int iPortNum, int iTimeout, BOOL bProtocol)**

: This function opens the printer communications resource.

Parameter

sIP_ADDR

Through the AP to set the IP address assigned to the printer.

iPortNum

Set the Port number. (default : 1470)

iTimeout

Set the connection time.

bProtocol

Specifies if protocol mode is enabled. If this member is TRUE, protocol mode

n **BOOL ClosePrinter()**

: Close the opened communication port.

- **Return value**

If the closing is in success, it will return the TRUE,

if the closing is failed, it will return the FALSE

n **int ControlCommand(BYTE *Cmd, int Length)**

: The function storing the printer control commands and string in the output buffer.

- **Parameter**

Cmds

BYTE type pointer sending the printer control command and string.

iLength

The number of byte of the Printer control command and string.

- **Return value**

If the saving is successful,

it will return the number of data byte stored in the output buffer.

If it exceed capacity of output buffer, it will return the **BUFFER_OVERFLOW**,

If the communication port is not opened, it will return the **NOT_OPEN_THE_PORT**.

n **int SaveSpool(TCHAR* sData, int iFontValue, BOOL bEmphasis)**

: The function storing the print data to the output buffer,

- **Parameter**

  sData

  Sending the print letter by the CString type

  Printer Control Command which does not contain NULL price has nothing to do although it includes at the character string.

  iFontValue

  Setting the font size of printed CString.

  0 = Normal -Default-

  1 = Double Height

  2 = Double Width

  3 = Double Height and Width

  bEmphasis

  Setting the emphasis mode

  FALSE = Normal

  TRUE = Emphasized

- **Return value**

  If the storing is successful,

  it will return the number of data byte stored in the output buffer.

  If it exceed capacity of output buffer, it will return the **BUFFER_OVERFLOW**,

  If the communication port is not opened, it will return the **NOT_OPEN_THE_PORT**.

n **int PrintSpool(BOOL Delete_Spool = TRUE)**

: printing the stored data in the output buffer to the communication port.

- **Parameter**

  bDelete_Spool

  If it is set to TRUE, delete the text of buffer clearly after printing,

  If it is set to FALSE, don't delete the text of buffer after printing.

- **Return value**

  If the printing is completed successfully,

  It will be returned the SUCCESS .

  The printing is not completed successfully because the communication port is not opened, it will be returned the NOT_OPEN_THE_PORT.

  A third parameter value of InitPrint function is FALSE, output an ETX message.

It outputs an PAPER_END message if a printing while paper falls off.

n **void Clear_Spool()**

: Deleting the text in buffer clearly.

n **void CardRead (bool Track2 = TRUE)**

: Sending the card reading command to the printer.

- **Parameter**

  Track2

  Track 2 is set to FALSE with bool type, the printer will be preparing for reading the card in Track2,

  Track 3 is set to TRUE, the printer will be preparing for reading the card in Track 3

n **void CardCancel()**

: Use when exit from card reading command.

n **int BmpSaveSpool(TCHAR* sBmpFilePath, int iXPosition, int iYPostion)**

: The function is convert BMP file which sends to the printer and to store at an output buffer

- **Parameter**

  sBmpFilePath

  BMP file is a process and name

  iXPosition

  BMP file to printer, the X-axis position output

  iYPosition

  BMP file to printer, the Y-axis position output

- **Return Value**

  If the storing is successful,

  it will return the number of data byte stored in the output buffer.

  If it exceed capacity of output buffer, it will return the **BUFFER_OVERFLOW**,

  If the communication port is not opened, it will return the **NOT_OPEN_THE_PORT**.

  If BMP file is not loaded, it will return the **-1** value.

n **int BarcodeSaveSpool (BYTE ucBarcodeType, BYTE ucBarWidth, BYTE ucBarHeight, BOOL HRI_On, TCHAR* BarcodeData, int iPrtWidth,int iPrtHeight,int iXPos);**

: Print barcode function.

- **Parameter**

ucBarcodeType

Select the type of barcode.

UPCA : 65(decimal)

UPCE : 66

EAN13 : 67

EAN8 : 68

CODE39 : 69

ITF : 70

CODABAR: 71

CODE93 : 72

CODE128: 73

ucBarWidth

Barcodes Width (Default: 2 Range: 2 <= m <= 6)

ucBarHeight

Set barcode height (dots per 1dot = 0.0125Cm)

HRI_On

HRI character output (TRUE: ON FALSE: OFF)

BarcodeData

Barcode data input

iPrtWidth

Print Width (_2INCH, _3INCH, _4INCH)

iPrtHeight

Print height (dots per set)

iXPos

X-axis position barcode output

- **Return value**

1 fixed.

6 WM_RECEIVE_MSG Message reference

Register "MSG_RECEIVE_CHECK" with using DefWindowProc function and make function receiving the message.

Ex> DefWindowProc(UINT message, WPARAM wParam, LPARAM lParam)

n In case of receiving the card data from the printer

Transmit the DATA to the WParam and transmit the card data to the LParam in BYTE type pointer.

n **When the printer receive the wrong data or misread the card.**

Transmit the NACK to the WParam and transmit the integer 0 to the LParam

n **When the printer receive the data or commandnormally**

Transmit the EOT to the WParam and tansmit the integer 0 to the LParam.

n **When the printer completed to print normally**

Transmit the ETX to the WParam, transmit the integer 0 to the LParam.