PRINTRONIX AUTO ID SDK for C# 1.0.0.1 (180906)

Purpose: To write & read RFID (This function only supports network & USB printers)

|  |  |  |
| --- | --- | --- |
| Sequence | SDK Function name | Description |
| 1 | openport(string ipaddress, int port)  return 0: success  return 1: fail | Open the network printer |
| openport()  return 0: success  return 1: fail | Open the USB printer |
| 2 | PTXA\_DLLinit() | To initialize DLL |
| option | PTXA\_FormSetup(int mode, int value)  mode:Length=1  mode:Count=2 | Set parameters for CREATE and EXECUTE command |
| 3 | PTXA\_SetupRFID(int length, bool incremental, string format, int increment\_amount, string data)  format B: binary  format D: decimal  format H: hexadecimal  return 0: success  return 1: fail | Set parameters for WriteRFID |
| 4 | PTXA\_WriteRFID(string LOCK, string membank)  LOCK null: without LOCK  membank EPC: EPC  membank TID: TID  membank USR: USR  membank ACS: ACS  membank KIL: KIL  membank PC: PC  return 0: success  return 1: fail | Program an RFID tag |
| 5 | PTXA\_VerifyRFID(string LOCK, string membank, int length, string format)  UNLOCK null: without UNLOCK  membank EPC: EPC  membank TID: TID  membank USR: USR  membank ACS: ACS  membank KIL: KIL  membank PC: PC  return 0: success  return 1: fail | To read the content of an RFID tag and return to the host |
| 6 | PTXA\_PrintForm(string filename)  return 0: success  return 1: fail | End the data form and print it |
| 7 | closeport()  return 0: success  return 1: fail | Close the opened printer |

Purpose: To request the printer status (This function only supports network & USB printers)

|  |  |  |
| --- | --- | --- |
| Sequence | SDK Function name | Description |
| 1 | openport(string ipaddress, int port)  return 0: success  return 1: fail | Open the network printer |
| openport()  return 0: success  return 1: fail | Open the USB printer |
| 2 | PTXA\_SetupStatus(int mode)  mode 1: Snoop PAR  mode 2: Snoop ETH  mode 3: Snoop SER  return 0: success  return 1: fail | Enable the snooper.  The printer status command will take effect when a snooper is enabled. |
| 3 | PTXA\_GetStatus(int delay\_millisecond)  delay\_millisecond: waiting for the status response delay (millisecond).  return 0: success  return 1: fail | Request to get printer status |
| 4 | PTXA\_SetupStatus(0)  mode 0: Snoop Off  return 0: success  return 1: fail | Dnable the snooper. |
| 5 | closeport()  return 0: success  return 1: fail | Close the opened printer |

Purpose: To send print job

|  |  |  |
| --- | --- | --- |
| Sequence | SDK Function name | Description |
| 1 | openport(string DriverPrinterName)  return 0: success  return 1: fail | Open the driver printer |
| openport(string ipaddress, int port)  return 0: success  return 1: fail | Open the network printer |
| openport()  return 0: success  return 1: fail | Open the USB printer |
| 2 | PTXA\_DLLinit() | To initialize DLL |
| option | PTXA\_WindowsFontImage(string logo\_name, string text, Font font)  return 0: success  return 1: fail | Creating a logo from windows font.  Must be first one after PTXA\_DLLinit if you want to use this function |
| option | PTXA\_Scale(int hdpi, int vdpi)  return 0: success  return 1: fail | Set the scale of the printer |
| option | PTXA\_JOBSetup(int mode, int value)  mode:Reset=0  mode:Width=1, 100<value<8500 (1/1000 inches)  mode:Length=2, 100<value<99000 (1/1000 inches)  mode:Speed=3, 1<value<10 (IPS)  mode:Intensity=4, -15<value<15  mode:Label=5, 0<value<4 (0:no sensor, 1:mark, 2:gap, 3:advanced gap, 4:advanced notch)  mode:Media=6, 0<value<4 (0:continuous mode, 1:tear-off scrip, 2:tear-off, 3:peel-off, 4:cut)  mode:Rotate=7, value=0, 90, 180, 270  return 0: success  return 1: fail | Set some print parameters |
| option | PTXA\_FontSetup(int mode, string value)  mode:Font Name=0, value=font name  mode:Font Bold=1, value="ON", "OFF"  mode:Font Slant=2, value="R", "L"  mode:Font Encoding=3, value="ASCII", "GB", "BIG5", "KSC", "SJIS", "UTF8"  return 0: success  return 1: fail | Set the font related parameters |
| option | PTXA\_FormSetup(int mode, int value)  mode:Length=1  mode:Count=2 | Set parameters for CREATE and EXECUTE command |
| 3 | PTXA\_CreateForm(string FormName)  return 0: success  return 1: fail | Create a data form |
| option | PTXA\_SetBarcode(int BcdType, int mode, string value)  BcdType:C39=0  BcdType:C128A=1  BcdType:C128B=2  BcdType:C128C=3  BcdType:EAN13=4  BcdType:UPCA=5  BcdType:MAXICODE=6  BcdType:QR=7  BcdType:PDF417=8  BcdType:AZTEC=9  BcdType:DMATRIX=10  BcdType:EAN8=11  BcdType:I25=12  BcdType:CODABAR=13  BcdType:UCC128=14  mode:RESET=0  mode:HEIGHT=1  mode:ROTATE=2  mode:MAGNIFY=3  mode:PDF=4  mode:XWIDTH=5  return 0: success  return 1: fail | Set barcode parameter |
| option | PTXA\_PrintBarcode(int BcdType, int SR, int SC, string data)  BcdType:C39=0  BcdType:C128A=1  BcdType:C128B=2  BcdType:C128C=3  BcdType:EAN13=4  BcdType:UPCA=5  BcdType:MAXICODE=6  BcdType:QR=7  BcdType:PDF417=8  BcdType:AZTEC=9  BcdType:DMATRIX=10  BcdType:EAN8=11  BcdType:I25=12  BcdType:CODABAR=13  BcdType:UCC128=14  SR:Staring row (dot)  SC:Starting column (dot)  return 0: success  return 1: fail | Print barcode |
| option | PTXA\_PrintTextEng(int SR, int SC, int VE, int HE, string data)  SR:Staring row (dot)  SC:Starting column (dot)  VE:Vertical expansion  HE:Horizontal expansion  return 0: success  return 1: fail | Print alphanumeric data |
| option | PTXA\_PrintTextCH(int SR, int SC, int VE, int HE, string data)  SR:Staring row (dot)  SC:Starting column (dot)  VE:Vertical expansion  HE:Horizontal expansion  return 0: success  return 1: fail | Print DBCS charactersdata |
| option | PTXA\_DrawBox(int LT, int SR, int SC, int ER, int EC)  LT:Line thickness (dot)  SR:Staring row (dot)  SC:Starting column (dot)  ER:Ending row (dot)  EC:Ending column (dot)  return 0: success  return 1: fail | Print rectangular boxes. |
| option | PTXA\_DrawLine(int LT, int SR, int SC, int ER, int EC)  LT:Line thickness (dot)  SR:Staring row (dot)  SC:Starting column (dot)  ER:Ending row (dot)  EC:Ending column (dot)  return 0: success  return 1: fail | Print lines |
| option | PTXA\_DrawCircle(int LT, int SR, int SC, int DA)  LT:Line thickness (dot)  SR:Staring row (dot)  SC:Starting column (dot)  DA:Diameter (dot)  return 0: success  return 1: fail | Print circles |
| option | PTXA\_PrintImage(int SR, int SC, string logoname)  SR:Staring row (dot)  SC:Starting column (dot)  return 0: success  return 1: fail | Print image |
| 4 | PTXA\_PrintForm(string filename)  return 0: success  return 1: fail | End the data form and print it |
| option | PTXA\_DeleteImage(string logo\_name)  return 0: success  return 1: fail | Delete the created logo |
| 5 | closeport()  return 0: success  return 1: fail | Close the opened printer |

Purpose: To send print job manually

|  |  |  |
| --- | --- | --- |
| Sequence | SDK Function name | Description |
| 1 | openport(string DriverPrinterName)  return 0: success  return 1: fail | Open the driver printer |
| openport(string ipaddress, int port)  return 0: success  return 1: fail | Open the network printer |
| openport()  return 0: success  return 1: fail | Open the USB printer |
| 2 | PTXA\_SendCmd(string command)  return 0: success  return 1: fail | Send the command to printer manually |
| 3 | closeport()  return 0: success  return 1: fail | Close the opened printer |