

[Extend Linux SDK]

[Printer Extend Command Development Manual v1.4]

1. Information of the Manual	3
2. Operation System	3
3. Remark	3
4. Method	4
4.1 ScanPos	4
4.2 ScanLabel	6
4.3 GetPrinterTaskOverInfo	8
4.4 SetPrintWidth	10
4.5 GetPrintWidth	11

1. Information of the Manual

This SDK manual provides the dll file information for Windows application development.

We continuously promote and update the function and quality of all our products. Any change to the product specification and the manual will be without any further notice.

2. Operation System

- Windows 7/8/10

3. Remark

- When error code Return Value is greater than 0, it is the internal error of Windows system, please refer to related help file.

4. Method

4.1 ScanPos

This function is used to broadcast UDP port 3298 to obtain information such as LAN printer IP addresses.

```
int ScanPos(  
  
    int timeout,  
  
    char* buf,  
  
    int buflen,  
  
    const char* UDPBroadCastIPAddress  
  
);
```

Parameter:

int timeout

[in] Set broadcast timeout period.

char buf*

[in,out]Obtained printer information. '|' is the symbol for dividing multiple printers, ';' is the sequential information for each printer:

1. IP address
2. Printer status
3. Printing width
4. SN number
- 5.MAC
6. Printer Name
7. Firmware version number
- 8.dpi
9. Network type
- 10.DHCP
11. Mask
12. Gateway
13. WiFi mode (WiFi mode)
14. WiFi Name (WiFi Mode)
15. Encryption method (WiFi mode)
16. Password. (WiFi mode)

Wifi encryption method enumeration:

0 "OPEN"

1: "WEP"

2: "WPA2-PSK"

3: "WPA/WPA2-PSK"

4: "WPA-PSK"

5: "WPA"

6: "WPA2"

7: "SAE"

8: "WPA2/WPA3-PSK"

*int * buflen*

[in]Buf lenth.

const char UDPBroadcastIPAddress*

[in]UDP broadcast address.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_DRIVER_TCP_NOT_FOUND	-103	TCP did not find the printer
HPRT_E_NOT_ENOUGH_BUFFER	-2	Not Enough buf

4.2 ScanLabel

This function is used to broadcast UDP port 6000 to obtain information such as LAN printer IP addresses.

```
int ScanLabel(  
    int timeout,  
    char* buf,  
    int buflen,  
    const char* UDPBroadCastIPAddress  
);
```

Parameter:

int timeout

[in] Set broadcast timeout period.

char buf*

[in,out]Obtained printer information. '|' is the symbol for dividing multiple printers, ';' is the sequential information for each printer:

1. IP address
2. Printer status
3. Printing width
4. SN number
- 5.MAC
6. Printer Name
7. Firmware version number
- 8.dpi
9. Network type
- 10.DHCP
11. Mask
12. Gateway
13. WiFi mode (WiFi mode)
14. WiFi Name (WiFi Mode)
15. Encryption method (WiFi mode)
16. Password. (WiFi mode)

Wifi encryption method enumeration:

0 "OPEN"

1: "WEP"

2: "WPA2-PSK"

3: "WPA/WPA2-PSK"

4: "WPA-PSK"

5: "WPA"

6: "WPA2"

7: "SAE"

8: "WPA2/WPA3-PSK"

*int * buflen*

[in]Buf lenth.

const char UDPBroadCastIPAddress*

[in]UDP broadcast address.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_DRIVER_TCP_NOT_FOUND	-103	TCP did not find the printer
HPRT_E_NOT_ENOUGH_BUFFER	-2	Not Enough buf

4.3 GetPrinterTaskOverInfo

This function is used to obtain information about the completion of the printer printing task (which needs to be used after PortOpen the printer, and for any operations that need to be read, please read it before calling this interface).

int GetPrinterTaskOverInfo(

void* handle,

int timeout,

char* buf,

int buflen,

int* returnDataNum

);

Parameter:

void handle*

[in,out] Target printer object created.

int timeout

[in] Set query timeout period.

char buf*

[in,out]Obtained printer status completion information. Including ID, num_index: page number printed, num_total: total number of copies printed, copy_index: current number of copies printed, copy_total: total number of copies printed, type: print completion status:

01001	job_print_start	Print
01002	job_print_ok	Homework printing completed
01003	job_print_fail	Homework printing failed
01004	job_pause	Homework pause
01010	job_cancel	Homework cancellation
01011	job_cancel_fatal	Exception caused homework cancellation
01012	job_cancel_button	Cancel button assignment
01013	job_cancel_app	Cancel the homework on the upper computer

*int * buflen*

[in]The length of buf.

*int * returnDataNum*

[in]Return the byte length to buf.

Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	No enough memory
E_INVALID_MODEL_TYPE	-3	This model does not support this feature.
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

4.4 SetPrintWidth

This function is used to set the printing width of the printer (can be saved when power is off, and needs to be used after PortOpen the printer).

int SetPrintWidth(

void* handle,

int widthDots

);

Parameter:

void handle*

[in,out] Target printer object created.

int widthDots

[in] Set the printing width dots.

Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	No enough memory
E_INVALID_MODEL_TYPE	-3	This model does not support this feature.
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

4.5 GetPrintWidth

This function is used to obtain the print width of the printer (needs to be used after PortOpen printer).

```
int GetPrintWidth(  
  
    void* handle,  
  
    Int* getWidthDots  
  
);
```

Parameter:

void handle*

[in,out] Target printer object created.

Int getWidthDots*

[in] Get print width in dots.

Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	No enough memory
E_INVALID_MODEL_TYPE	-3	This model does not support this feature.
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout