SYNO API for Linux(UNIX) User Manual

V1.2

Hangzhou Grow Technology Co., Ltd
www.hzgrow.com

1. API Function Description

1.1 function

1. OpenDevice;
BOOL WINAPI PSOpenDevice(int nDeviceType,int nPortNum,int nPortPara,int nPackageSize=2);
Parameter: nDeviceType: 0:USB Device ;1:Serial Port Device
nPortNum: if nDeviceType equal 1 then nPortNum is a Serail port number representation.
nPortPara: baudRate setting
Return: if device opened success then the function return true.

2. CloseDevice
BOOL WINAPI PSCloseDevice();

3. Detect finger and Get Image
int WINAPI PSGetImage(int nAddr);
Return: if device closed success then the function return true.

4. Upload Original Image
int WINAPI PSUpImage(int nAddr,[out]unsigned char* pImageData, [out]int* iImageLength);
Parameter: pImageData: image data buffer.the buffer size is 256*288,the width of image is 256,the height of image is 288.each of data of array is the gray one pixels,the gray range from 0 t0 255.
iImageLength:the image size.
Return: function return 0 Indicates success,others see error code reference.

5. Generate Character file
int WINAPI PSGenChar(int nAddr,[in] int iBufferID);
Parameter: iBufferID: Indicates a character file buffer chip internal. The value is always 0 or 1.
Return: function return 0 Indicates success,others

6. Match two character file on chip
int WINAPI PSMatch(int nAddr,[out] int* iScore);
Parameter: iScore: if the function return success then this variable can get a Score of one Matching.
Return: function return 0 Indicates success,others see error code reference.

7. Search a part or all of fingerprint libray
int WINAPI PSSearch(int nAddr,int iBufferID, int iStartPage, int iPageNum, [out]int *iMbAddress);
Parameter: iBufferID: Indicates a character file buffer chip internal. The
value is always 0 or 1.
iStartPage:Indicates a Start Page of search.
iPageNum: Indicates numbers of Page to search.
iMbAddress: if find a valid character ,the variable retrieve page
id
Return: function return 0 Indicates success,others see error code
reference.

8. Combine BufferA's character file with BufferB's character file and generate the templet

int WINAPI PSRegModule(int nAddr);

Return: function return 0 Indicates success,others see error code
reference.

9. Store BufferA or BufferB's character file to flash fingerprint library
int WINAPI PSStoreChar(int nAddr,int iBufferID, int iPageID); Parameter: iBufferID: Indicates a character file buffer chip internal. The
value is always 0 or 1.
iPageID: this Indicates character file storage page.from 0 to 255 Return: function return 0 Indicates success,others see error code
reference.

10. Transfer a templet to BufferA or BufferB from flash fingerprint library
int WINAPI PSLoadChar(int nAddr,int iBufferID,int iPageID);
Parameter: same as PSStoreChar.
Return: function return 0 Indicates success,others see error code
reference.

11. Transfer character file from BufferA or BufferB to PC
int WINAPI PSUpChar(int nAddr,int iBufferID, [out]unsigned char* pTemplet, [out]int* iTempletLength);
Parameter: iBufferID :same as PSStoreChar.
* pTemplet: Receive character file data from specify buffer.
iTempletLength: serial port communicate used. Indicates
received datas length.
Return: function return 0 Indicates success,others see error code
reference.

12. Download a character form form pc to BufferA or BufferB
int WINAPI PSDownChar(int nAddr,int iBufferID,[in] unsigned char* pTemplet, [in]int iTempletLength);
Parameter: same as PSUpChar.

Return: function return 0 Indicates success,others see error code
reference.

13. Delete specify range of character file from flash fingerprint libaray
int WINAPI PSDelChar(int nAddr,int iStartPageID,int nDelPageNum); Parameter: iStartPageID: start of range.
nDelPageNum: numbers of character file.
Return: function return 0 Indicates success,others see error code
reference.

14. Clear flash fingerprint libaray
int WINAPI PSEmpty(int nAddr);
Return: function return 0 Indicates success,others see error code
reference.

15. Verify Device Communicate Key
int WINAPI PSVfyPwd(int nAddr,unsigned char* pPassword);

16. Read Notepad
int WINAPI PSReadInfo(int nAddr,int nPage,[out]unsigned char* UserContent);
Parameter: nPage: specify page of notepad.
UserContent: the content buffer.
Return: function return 0 Indicates success,others see error code
reference.

17. Write Notepad
int WINAPI PSWriteInfo(int nAddr,int nPage,[in]unsigned char* UserContent);
Parameter: nPage: specify page of notepad.
UserContent: the content buffer.
Return: function return 0 Indicates success,others see error code
reference.

18. Set baudrate
int WINAPI PSSetBaud(int nAddr,int nBaudNum);

Parameter: nBaudNum: set baudrate.
Return: function return 0 Indicates success,others see error code
reference.

19. Set security level
int WINAPI PSSetSecurLevel(int nAddr,int nLevel);
Parameter: nBaudNum: set Security Level.
Return: function return 0 Indicates success,others see error code
reference.

20. Get random data generate by chip
int WINAPI PSGetRandomData(int nAddr,unsigned char* pRandom); Parameter: pRandom: Random data buffer.

Return: function return 0 Indicates success,others see error code
reference.
21. Format error information
char* WINAPI PSErr2Str(int nErrCode);
Parameter: nErrCode: Error code.

Return: function return a detail information of corresponding error code

4.3 Error code

```
#define PS_OK              0x00
#define PS_COMM_ERR        0x01
#define PS_NO_FINGER       0x02
#define PS_GET_IMG_ERR     0x03
#define PS_FP_TOO_DRY      0x04
#define PS_FP_TOO_WET      0x05
#define PS_FP_DISORDER     0x06
#define PS_LITTLE_FEATURE  0x07
#define PS_NOT_MATCH       0x08
#define PS_NOT_SEARCHED    0x09
#define PS_MERGE_ERR       0x0a
#define PS_ADDRESS_OVER    0x0b
#define PS_READ_ERR        0x0c
#define PS_UP_TEMP_ERR     0x0d
#define PS_RECV_ERR        0x0e
#define PS_UP_IMG_ERR      0x0f
#define PS_DEL_TEMP_ERR    0x10

#define PS_CLEAR_TEMP_ERR 0x11
#define PS_SLEEP_ERR 0x12
#define PS_INVALID_PASSWORD 0x13
#define PS_RESET_ERR 0x14
#define PS_INVALID_IMAGE 0x15
#define PS_HANGOVER_UNREMOVE 0X17
```