Doc. #: SS-19-0719-04
Date: 13, Jul. 2021
Ver.2.4

LCOS-SLM

Liquid Crystal Based Spatial Light Modulator

Programmer’s Guide

This document provides the application programming interface (API) for the SLMFunc.DLL function library.

Notes to Users

2) Information in this Operation Manual is subject to change without notice.

3) Information of this Operation Manual is prepared with careful examination, however, in the event of any mistake, please contact us.

Notes in Bringing This Product Out of Japan

1) When this product is brought out of Japan, some laws or regulations of a destination country may prohibit this product from being used there. In such countries, the use of this product may lead to punishment. Please note, that in such cases Santec Corporation shall not be held responsible in any way.

2) When this product is exported (or brought out of Japan), this product is applicable to a strategic material specified in the "Foreign Exchange and Foreign Trade Control Law", then under law of the Japanese Government, an export permit is required.

Contents

santec

1 Introduction

1.1 Description

Santec provides DLL application interface for SLM control drivers.

This document provides application programming interface (API) for function library.

1.2 Function block diagram

This product consists of LCOS unit, Drive board and Option board.

The function of each part is as follows.

<Option board>

Convert input data from each interface to data format of Drive board.

<Drive board>

Display converted data on LCOS unit.

<LCOS unit>

Display phase pattern.

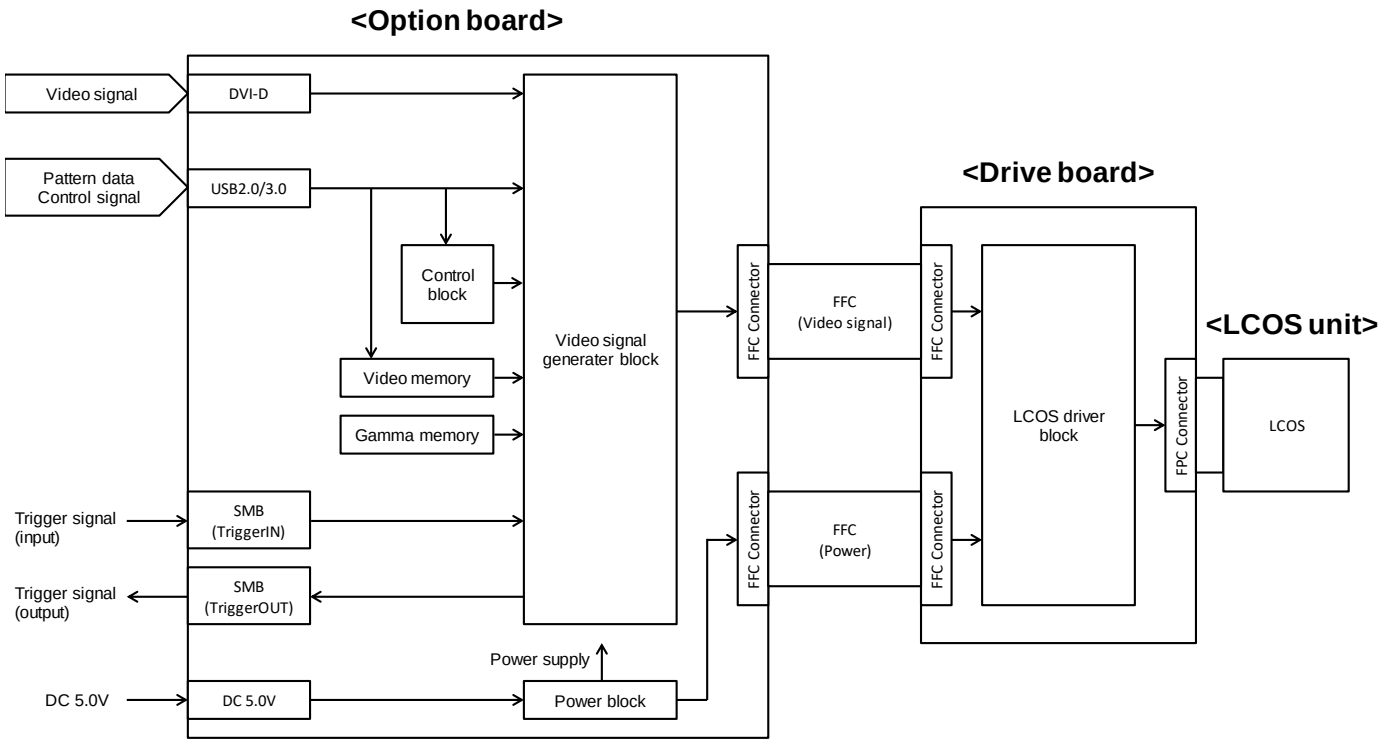
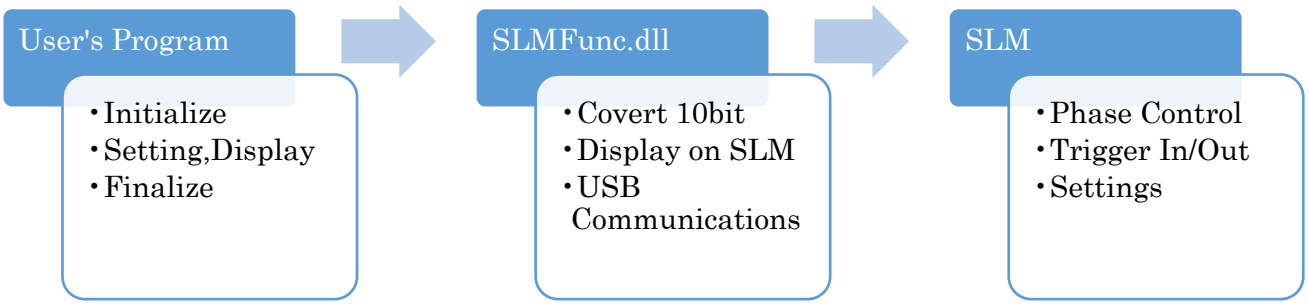


Fig. 1.2-1 Function block diagram

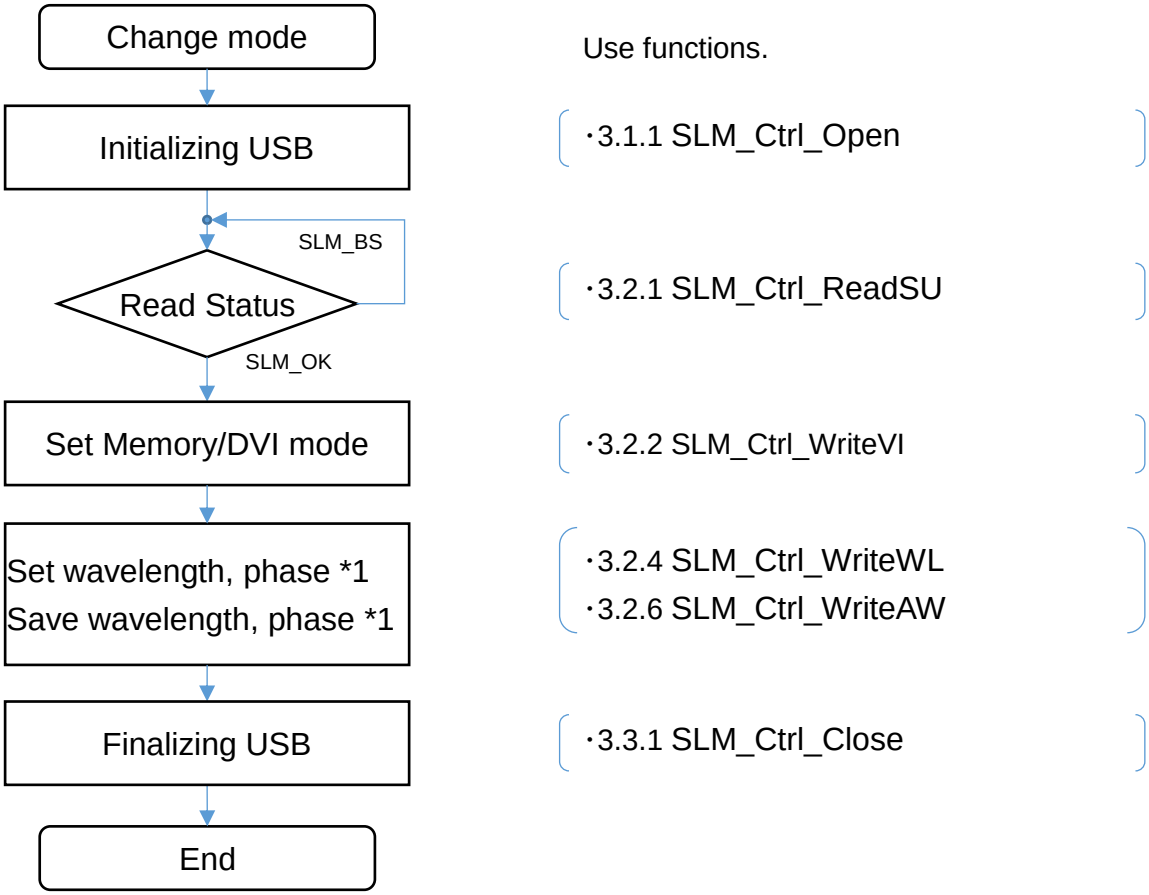
1.3 Process Flow

Place the SLMFunc.dll in the same folder as user program, and display data by calling SLM function in user's program.



1.3.1 Set mode, wavelength, phase

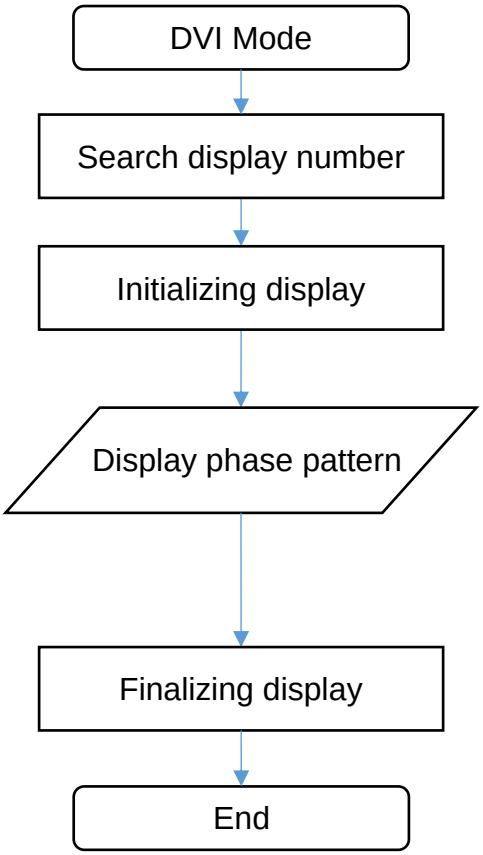
SLM has two modes, which can be changed by functions.
And, the wavelength and phase need to be set only once.



*1 Once saved, there is no need to set them each time.

1.3.2 DVI mode

In DVI mode, display on LCOS by DVI input.



Use functions.

- 2.4.1 SLM_Disp_Info
- 2.4.2 SLM_Disp_Info2

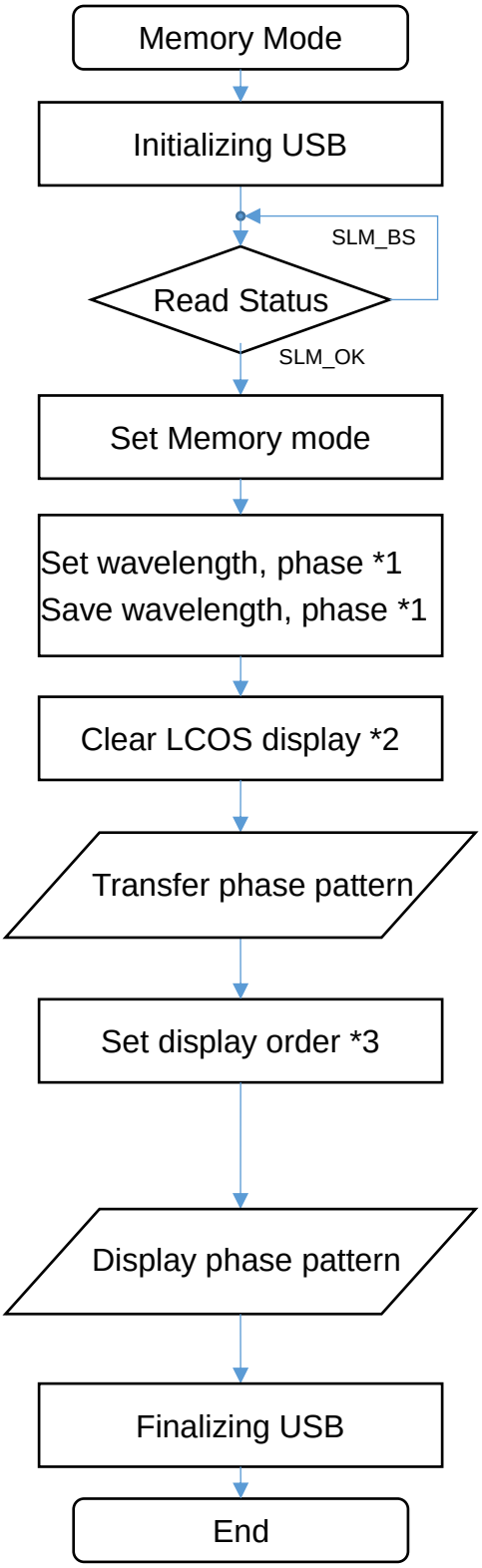
- 2.1.1 SLM_Disp_Open

- 2.2.1 SLM_Disp_GrayScale
- 2.2.2 SLM_Disp_BMP
- 2.2.3 SLM_Disp_Data
- 2.2.4 SLM_Disp_ReadBMP
- 2.2.5 SLM_Disp_ReadBMP_A
- 2.2.6 SLM_Disp_ReadCSV
- 2.2.7 SLM_Disp_ReadCSV_A

- 2.3.1 SLM_Disp_Close

1.3.3 Memory Mode

In the memory mode, the phase data is transferred to the memory to SLM and displayed on LCOS by specifying the memory number.



Use functions.

- ・3.1.1 SLM_Ctrl_Open
- ・3.2.1 SLM_Ctrl_ReadSU
- ・3.2.2 SLM_Ctrl_WriteVI
- ・3.2.4 SLM_Ctrl_WriteWL
- ・3.2.6 SLM_Ctrl_WriteAW
- ・3.2.7 SLM_Ctrl_WriteGS
- ・3.2.10 SLM_Ctrl_WriteMI
- ・3.2.11 SLM_Ctrl_WriteMI_BMP
- ・3.2.12 SLM_Ctrl_WriteMI_BMP_A
- ・3.2.13 SLM_Ctrl_WriteMI_CSV
- ・3.2.14 SLM_Ctrl_WriteMI_CSV_A
- ・3.2.16 SLM_Ctrl_WriteMT
- ・3.2.18 SLM_Ctrl_WriteMR
- ・3.2.20 SLM_Ctrl_WriteMP
- ・3.2.21 SLM_Ctrl_WriteMZ
- ・3.2.22 SLM_Ctrl_WriteMW
- ・3.2.32 SLM_Ctrl_WriteTC
- Select display
  - ・3.2.24 SLM_Ctrl_WriteDS
- Continuous display
  - ・3.2.22 SLM_Ctrl_WriteMW
  - ・3.2.26 SLM_Ctrl_WriteDR
  - ・3.2.27 SLM_Ctrl_WriteDB
- Trigger display
  - ・Trigger Input or 3.2.34 SLM_Ctrl_WriteTS
- ・3.3.1 SLM_Ctrl_Close

*1 Once saved, there is no need to set them each time.
*2. Clear the display because SLM_Ctrl_MI** command cannot be used to write to the displayed Memory number.
*3 Not required when using SLM_Ctrl_WriteDS function.

1.4 Attention

1.4.1 Display number

When SLM is connected to a notebook computer, it is recognized as Display 2.
In the case of desktop computer, you need to check what display SLM recognizes.
If there are more than two displays, check the SLM's display number with
” 2.4.2 SLM_Disp_Info2” function.

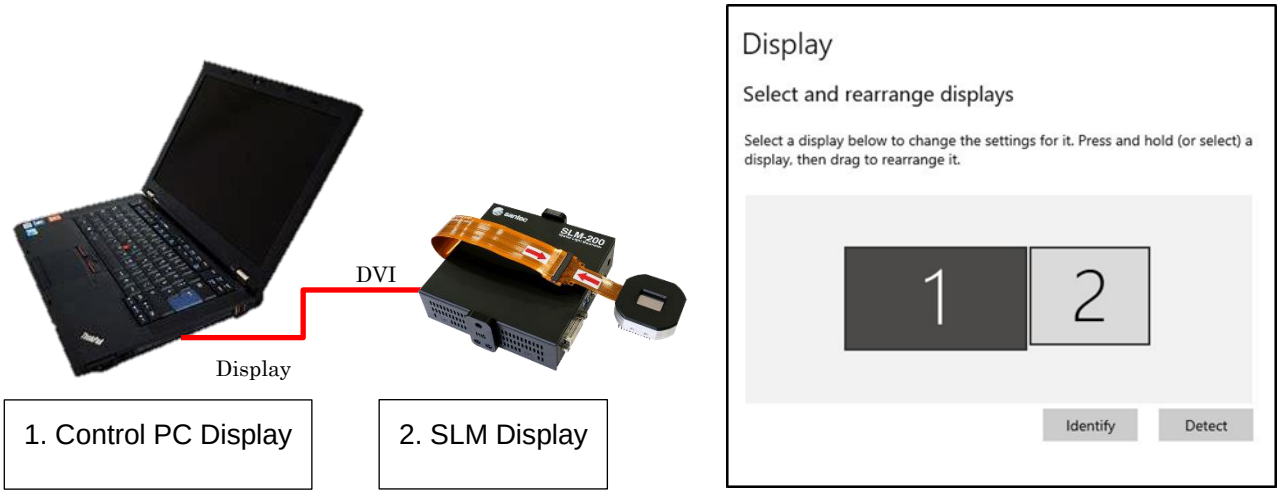
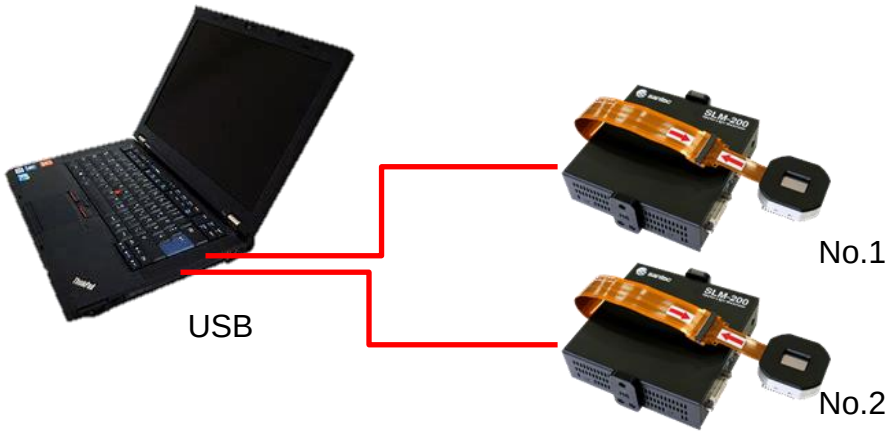


Figure 1.4-1 Display number

1.4.2 SLM Number

SLM Number is automatically allocated by Windows.



1.4.3 Supported OS

Windows 10

1.4.4 Development environment

Recommended development environment:

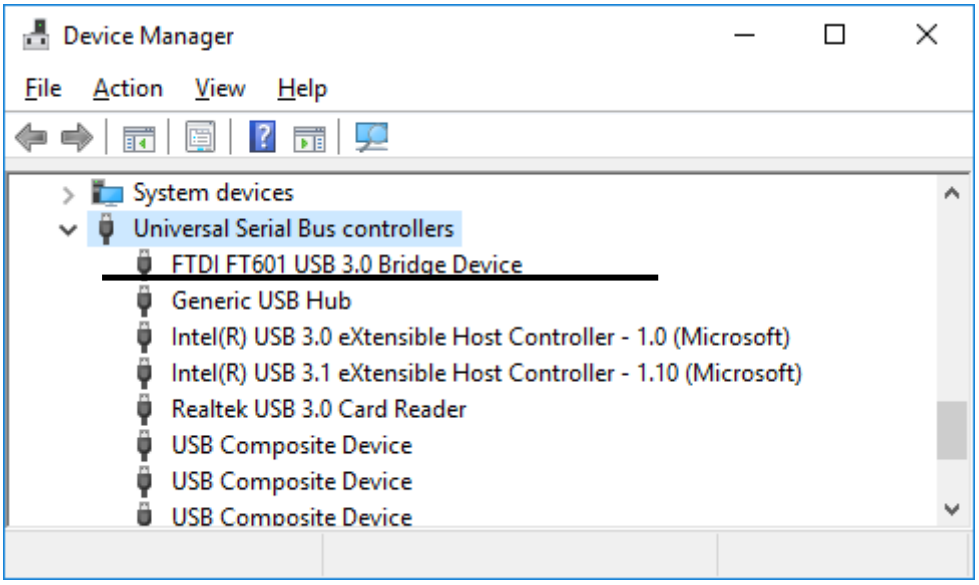
- Visual Studio
- Python
- MATLAB
- LabVIEW

Select a DLL according to 64-bit and 32-bit development environment.

¥x64¥SLMFunc.dll	・・・	64bit development environment.
¥x64¥FTD3XX.dll	・・・	64bit development environment.
¥x86¥SLMFunc.dll	・・・	32bit development environment.
¥x86¥FTD3XX.dll	・・・	32bit development environment.

Place the SLMFunc.dll and FTD3XX.dl in the same folder as user program.

Check that the FT601 recognizes the following when connecting the PC and SLM via USB.



If the FT601 does not recognize, turn off the SLM or refer to “USB driver installation procedure” section in the “SLM-200 OPERATIONAL MANUAL”.

1.4.5 Available DLL functions

Table 1.4-1 Available DLL functions

	Functions	GUI Software	DLL Functions
DVI I/F	Display CSV or BMP file	✓	✓
	Display array data		✓
	Full screen contrast	✓	✓
USB I/F	Set wavelength	✓	✓
	Continuous display	✓	✓
	Pattern capture	✓	✓
	Set trigger	✓	✓
	Display mode select	✓	✓
Other	CGH generator	✓	

2 Display Functions

2.1 Initializing

2.1.1 SLM_Disp_Open

```
SLM_STATUS
SLM_Disp_Open(
   DWORD DisplayNumber
)
```

Summary

SLM display initializing.

Parameters

DisplayNumber: Specify display number (1, 2, 3…).

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// Open Display2
if(SLM_Disp_Open(2) == SLM_OK){
        // OK
}
else{
        // Error
}
```

2.2 Display

2.2.1 SLM_Disp_GrayScale

```
SLM_STATUS
   SLM_Disp_GrayScale(
   DWORD DisplayNumber,
   DWORD Flags,
   USHORT GrayScale
)
```

Summary

Drawing the entire display with GrayScale input.

Parameters

DisplayNumber:	Specify display number (1, 2, 3…).
Flags:	Use this to change the display method. (Refer to “3.5 BMP, CSV, Data Flags”) If you use 120Hz model, use FLAGS_RATE120.
GrayScale:	Specify grayscale from 0 to 1023 (0π - 2 π).

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// Display all pixel 512
if(SLM_Disp_GrayScale(2,0,512) == SLM_OK){
        // OK
}

// Display grayscale on the 120Hz SLM.
if(SLM_Disp_GrayScale(2, FLAG_RATE120,512) == SLM_OK){
        // OK
}
```

2.2.2 SLM_Disp_BMP

```
SLM_STATUS
SLM_Disp_BMP(
   DWORD DisplayNumber,
   DWORD Flags,
   HBITMAP bmp
)
```

Summary

Display BMP on the SLM.

Parameters

DisplayNumber:	Specify display number (1, 2, 3…).
Flags:	This parameter is for future option (default value 0).

Use this to change the display method. (Refer to “3.5 BMP, CSV, Data Flags”)
If you use 120Hz model, use FLAGS_RATE120.

bmp:	Pointer to bmp.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// Display bmp on the SLM.

HBITMAP hbmp;

hbmp=(HBITMAP)LoadImage(hinst,_T("c:¥¥test.bmp”),
IMAGE_BITMAP,0,0,LR_CREATEDIBSECTION | LR_LOADFROMFILE);

if(SLM_Disp_BMP(2,0,hbmp) == SLM_OK){
        // OK
}
```

2.2.3 SLM_Disp_Data

```
SLM_STATUS
SLM_Disp_Data(
   DWORD DisplayNumber,
   USHORT width,
   USHORT height,
   DWORD Flags,
   USHORT* data
)
```

Summary

Display array data on the SLM.

Parameters

DisplayNumber:	Specify display number (1, 2, 3…).
width:	Specify display width value.
height:	Specify display height value.
Flags:	Use this to change the display method. (Refer to “3.5 BMP, CSV, Data Flags”) If you use 120Hz model, use FLAGS_RATE120.
data:	Pointer to array of unsigned short data.(width * height * 2byte)

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// Display array data on the SLM.
USHORT *dat, *pos;
dat = pos = (USHORT*)malloc(sizeof(short) * 1920 * 1200);
for (int y = 0; y < 1200; y++) {
        for (int x = 0; x < 1920; x++) {
                *pos = (USHORT)(rand() * 1023);        // All data random
                pos++;
        }
}
if (SLM_Disp_Data(1, 1920, 1200, 0, dat) == SLM_OK) {
        // OK
}



// Display bmp on the 120Hz SLM.
if (SLM_Disp_Data(1, 1920, 1200, FLAG_RATE120, dat) == SLM_OK) {
        // OK
```

```
}
```

2.2.4 SLM_Disp_ReadBMP

```
SLM_STATUS
SLM_Disp_ReadBMP(
   DWORD DisplayNumber,
   DWORD BMPFlags,
   LPCWSTR FileName
)
```

Summary

Display bmpfile (Unicode) data on the SLM.

Parameters

DisplayNumber:	Specify display number (1, 2, 3…).
BMPFlags:	Use this to change the display method. (Refer to “3.5 BMP, CSV, Data Flags”) If you use 120Hz model, use FLAGS_RATE120.
FileName:	Pointer to buffer containing Unicode bmpfile name.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// bmp file
if (SLM_Disp_ReadBMP(2, 0, _T("C:¥¥test.bmp")) == SLM_OK) {
        // OK
}


// Display bmp on the 120Hz SLM.
if (SLM_Disp_ReadBMP(2, FLAG_RATE120, _T("C:¥¥test.bmp")) == SLM_OK) {
        // OK
}
```

2.2.5 SLM_Disp_ReadBMP_A

```
SLM_STATUS
SLM_Disp_ReadBMP_A(
   DWORD DisplayNumber,
   DWORD BMPFlags,
   LPCSTR FileName
)
```

Summary

Display bmpfile(ANSI code) data on the SLM.

Parameters

DisplayNumber:	Specify display number (1, 2, 3…).
BMPFlags:	Use this to change the display method. (Refer to “3.5 BMP, CSV, Data Flags”) If you use 120Hz model, use FLAGS_RATE120.
FileName:	Pointer to buffer containing ANSI code bmpfile name.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// bmp file
if (SLM_Disp_ReadBMP_A(2, 0, ("C:¥¥test.bmp")) == SLM_OK) {
        // OK
}


// Display bmp on the 120Hz SLM.
if (SLM_Disp_ReadBMP_A(2, FLAG_RATE120, ("C:¥¥test.bmp")) == SLM_OK) {
        // OK
}
```

2.2.6 SLM_Disp_ReadCSV

```
SLM_STATUS
SLM_Disp_ReadCSV(
   DWORD DisplayNumber,
   DWORD CSVFlags,
   LPCWSTR FileName
)
```

Summary

Display csvfile(Unicode) data on the SLM.

Parameters

DisplayNumber:	Specify display number (1, 2, 3…).
CSVFlags:	Use this to change the display method. (Refer to “3.5 BMP, CSV, Data Flags”) If you use 120Hz model, use FLAGS_RATE120.
FileName:	Pointer to buffer containing Unicode csvfile name. (Refer to “CSV Format3.6 CSV Format”)

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// CSV file
if (SLM_Disp_ReadCSV(2, 0, _T("C:¥¥test.csv”)) == SLM_OK) {
        // OK
}


// Display csv on the 120Hz SLM.
if (SLM_Disp_ReadCSV(2, FLAG_RATE120, _T("C:¥¥test.csv”)) == SLM_OK) {
        // OK
}
```

2.2.7 SLM_Disp_ReadCSV_A

```
SLM_STATUS
SLM_Disp_ReadCSV_A(
   DWORD DisplayNumber,
   DWORD CSVFlags,
   LPCSTR FileName
)
```

Summary

Display csvfile(ANSI code) data on the SLM.

Parameters

DisplayNumber:	Specify display number (1, 2, 3…).
CSVFlags:	Use this to change the display method. (Refer to “3.5 BMP, CSV, Data Flags”) If you use 120Hz model, use FLAGS_RATE120.
FileName:	Pointer to buffer containing ANSI code csvfile name. (Refer to “CSV Format3.6 CSV Format”)

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// csv file
if (SLM_Disp_ReadCSV_A (2, 0, ("C:¥¥test.csv”)) == SLM_OK) {
        // OK
}


// Display csv on the 120Hz SLM.
if (SLM_Disp_ReadCSV_A(2, FLAG_RATE120, ("C:¥¥test.csv”)) == SLM_OK) {
        // OK
}
```

2.3 SLM Finalizing

2.3.1 SLM_Disp_Close

```
SLM_STATUS
SLM_Disp_Close(
   DWORD DisplayNumber
)
```

Summary

SLM display finalizing.

Parameters

DisplayNumber: Specify display number (1, 2, 3…).

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// Close Display2
if(SLM_Disp_Close(2) == SLM_OK){
        // OK
}
```

2.4 Others

2.4.1 SLM_Disp_Info

```
SLM_STATUS
SLM_Disp_Info(
   DWORD DisplayNumber,
   USHORT *width,
   USHORT *height
)
```

Summary

Read width and height of the display.

Parameters

DisplayNumber:	Specify display number (1, 2, 3…).
width:	Pointer to unsigned short to store width value.
height:	Pointer to unsigned short to store height value.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// Display2 Information
UShort width, height;
if(SLM_Disp_Info(2, &width, &height) == SLM_OK){
        // OK
        // width = 1920, height = 1200
}
```

2.4.2 SLM_Disp_Info2

```
SLM_STATUS
SLM_Disp_Info(
   DWORD DisplayNumber,
   USHORT *width,
   USHORT *height,
   LPSTR DisplayName
)
```

Summary

Read width and height, DisplayName of display.

Parameters

DisplayNumber:	Specify display number (1, 2, 3…).
width:	Pointer to unsigned short to store width value.
height:	Pointer to unsigned short to store height value.
DisplayName	Pointer to a 128-byte buffer to store DisplayName. DisplayName format is “UserFriendlyName,ManufactreName, ProductCodeID,SerialNumberID” e.g. DisplayName = “LCOS-SLM,SOC,8001,2018021001”

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// Display2 Information
UShort width, height;
char DisplayName[128];

if(SLM_Disp_Info2(1, &width, &height, DisplayName) == SLM_OK){
        // OK
        // width = 1920, height = 1080
        // DisplayName = “ABC Z246,ABC,0001,ABC0123456789”      <= not SLM.
}
if(SLM_Disp_Info2(2, &width, &height, DisplayName) == SLM_OK){
        // OK
        // width = 1920, height = 1200
        // DisplayName = “LCOS-SLM,SOC,8001,2018021001”         <= SLM.
}
```

3 Control Functions

3.1 Initializing

3.1.1 SLM_Ctrl_Open

```
SLM_STATUS
SLM_Ctrl_Open (
   DWORD SLMNumber,
)
```

Summary

Open USB interface.

Parameters

SLMNumber: Specify SLM number (1-8).

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// Open USB interface
if(SLM_Ctrl_Open (1) == SLM_OK){
        // OK
}
```

3.2 Control

3.2.1 SLM_Ctrl_ReadSU

```
SLM_STATUS
SLM_Ctrl_ReadSU(
   DWORD SLMNumber
)
```

Summary

Read status of SLM. Busy or Ready.

Parameters

SLMNumber: Specify SLM number (1, 2, 3…).

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Notes

It will be in BUSY state for about 40 seconds for power ON and phase table expand.

Example

```
// Reads status
SLM_STATUS ret;

for(int i = 0; i < 100; i++){
        Sleep(1000);
        ret = SLM_Ctrl_ReadSU(1);
        if(ret == SLM_OK) return true;
        else if(ret == SLM_BS) continue;
        else return false;                    // error
}

return false;              // timeout
```

3.2.2 SLM_Ctrl_WriteVI

```
SLM_STATUS SLM_Ctrl_WriteVI(
   DWORD SLMNumber,
   DWORD mode
)
```

Summary

Write video mode DVI or Memory mode.

Parameters

SLMNumber:	Specify SLM number (1-8).
mode:	Specify mode value.
	0:Memory mode, 1:DVI mode, Default 1

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Notes

It takes 40 seconds to respond to expand phase table.

Example

```
// Set video mode 0
if(SLM_Ctrl_WriteVI (1,0) == SLM_OK){
        // OK
}
```

3.2.3 SLM_Ctrl_ReadVI

```
SLM_STATUS
SLM_Ctrl_ReadVI(
   DWORD SLMNumber,
   DWORD *mode
)
```

Summary

Read display mode DVI or Memory mode.

Parameters

SLMNumber:	Specify SLM number (1-8).
mode:	Pointer to unsigned int(32bit) to store mode value. 0: Memory mode, 1: DVI mode

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// Read display mode
DWORD mode;
if(SLM_Ctrl_ReadVI(1,&mode) == SLM_OK){
        // OK
}
else{
        // Error
}
```

3.2.4 SLM_Ctrl_WriteWL

```
SLM_STATUS
SLM_Ctrl_WriteWL(
   DWORD SLMNumber,
   DWORD wavelength,
   DWORD phase
)
```

Summary

Write wavelength and phase value.

It cannot be set to a value that causes internal calculation result of SLM to be abnormal.

e.g. Set phase 2.00π => calculation result 2.01π

Parameters

SLMNumber:	Specify SLM number (1-8).
wavelength:	Specify wavelength value.(e.g. 1500)
phase:	Specify phase value multiplied by 100 (0-999). e.g. 2.00π => 200.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Notes

It takes 40 seconds to respond to expand phase table.

Example

```
// Set wavelength (1500nm) and phase (2π)
if(SLM_Ctrl_WriteWL (1,1500,200) == SLM_OK){
        // OK
}
```

3.2.5 SLM_Ctrl_ReadWL

```
SLM_STATUS
SLM_Ctrl_ReadWL(
   DWORD SLMNumber,
   DWORD *wavelength,
   DWORD *phase
)
```

Summary

Read wavelength and phase value.

Parameters

SLMNumber:	Specify SLM number (1,2,3…8).
wavelength:	Pointer to unsigned int(32bit) to store wavelength value.(450-1600)
phase:	Pointer to unsigned int(32bit) to store phase value multiplied by 100. (0-999)

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// Read display mode
DWORD wavelength,phase;
if(SLM_Ctrl_ReadWL(1,&wavelength,&phase) == SLM_OK){
        // OK
        printf(“%d nm, %0.2f pai”, wavelength, ((float)phase)/100);
}
else{
        // Error
}
```

3.2.6 SLM_Ctrl_WriteAW

```
SLM_STATUS
SLM_Ctrl_WriteAW(
   DWORD SLMNumber
)
```

Summary

Save wavelength and phase settings.

The settings are retained even when power is turned off.

Parameters

SLMNumber: Specify SLM number (1-8).

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// Set video mode 0
SLM_Ctrl_WriteWL(1,1500,200);

if(SLM_Ctrl_WriteAW (1) == SLM_OK){
        // OK
}
```

3.2.7 SLM_Ctrl_WriteGS

```
SLM_STATUS
SLM_Ctrl_WriteGS(
   DWORD SLMNumber,
   USHORT GrayScale
)
```

Summary

Display specified grayscale on the entire display.

Parameters

SLMNumber:	Specify SLM number (1, 2, 3…).
GrayScale:	Specify grayscale from 0 to 1023 (0π - 2 π).

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// entire display 1023
if(SLM_Ctrl_WriteGS(1,1023) == SLM_OK){
        // OK
}
else{
        // Error
}
```

3.2.8 SLM_Ctrl_ReadGS

```
SLM_STATUS
SLM_Ctrl_ReadGS(
   DWORD SLMNumber,
   USHORT *GrayScale
)
```

Summary

Read grayscale on display.

Parameters

SLMNumber:	Specify SLM number (1, 2, 3…).
GrayScale:	Specify grayscale from 0 to 1023 (0π - 2 π).

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// read grayscale
USHORT grayscale;
if(SLM_Ctrl_ReadGS(1,&grayscale) == SLM_OK){
        // OK
}
else{
        // Error
}
```

3.2.9 SLM_Ctrl_WriteMC

```
SLM_STATUS
SLM_Ctrl_WriteMC(
   DWORD SLMNumber,
   DWORD MemoryNumber
)
```

Summary

Transfer phase pattern input from the DVI input to internal memory.

Parameters

SLMNumber:	Specify SLM number (1-8).
MemoryNumber:	Specify Memory number (1-128).

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// DVI input to internal memory 10.
if(SLM_Ctrl_WriteMC (1,10) == SLM_OK){
        // OK
}
```

3.2.10 SLM_Ctrl_WriteMI

```
SLM_STATUS
SLM_Ctrl_WriteMI(
   DWORD SLMNumber,
   DWORD MemoryNumber,
   USHORT width,
   USHORT height,
   DWORD Flags,
   USHORT* data
)
```

Summary

Transfer array data to SLM memory.

Parameters

SLMNumber:	Specify SLM number (1-8).
MemoryNumber:	Specify SLM number (1-128).
width:	Specify width value (1920).
height:	Specify SLM number (1200).
Flags:	This parameter is for future option (default value 0).
data:	Pointer to array of unsigned short data.

For detailed information about memory number refer to “3.7 Display table setting”.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Note

Since memory number displayed with SLM_Ctrl_WriteDS function cannot be overwritten, you need to change displayed memory number or change the displayed content with SLM_Ctrl_WriteGS function to write.

Example

```
// write array data to memory number 1
USHORT *dat, *pos;
dat = pos = (USHORT*)malloc(sizeof(short) * 1920 * 1200);
for (int y = 0; y < 1200; y++) {
        for (int x = 0; x < 1920; x++) {
                *pos = (USHORT)(rand() * 1023);        // All data random
                pos++;
        }
}
if (SLM_Ctrl_WriteMI(1, 1,1920, 1200, 0, dat) == SLM_OK) {   // OK}
```

3.2.11 SLM_Ctrl_WriteMI_BMP

```
SLM_STATUS
SLM_Ctrl_WriteMI_BMP(
   DWORD SLMNumber,
   DWORD MemoryNumber,
   DWORD Flags,
   LPCWSTR FileName
)
```

Summary

Transfer BMP file(Unicode) to SLM memory.

Parameters

DisplayNumber:	Specify display number (1, 2, 3…).
MemoryNumber:	Specify memory number(1-128).
BMPFlags:	Specify color mode. See “3.5 BMP, CSV, Data Flags”
FileName:	Pointer to buffer containing Unicode bmpfile name.

For detailed information about memory number refer to “3.7 Display table setting”.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Note

Since memory number displayed with SLM_Ctrl_WriteDS function cannot be overwritten, you need to change displayed memory number or change displayed content with SLM_Ctrl_WriteGS function to write.

Example

```
// bmp file to memory number 1
if (SLM_Ctrl_WriteMI_BMP(1,1, 0, _T("C:¥¥test.bmp”)) == SLM_OK) {
        // OK
}
```

3.2.12 SLM_Ctrl_WriteMI_BMP_A

```
SLM_STATUS
SLM_Ctrl_WriteMI_BMP_A(
   DWORD SLMNumber,
   DWORD MemoryNumber,
   DWORD Flags,
   LPCSTR FileName
)
```

Summary

Transfer BMP file(Unicode) to SLM memory.

Parameters

DisplayNumber:	Specify display number (1, 2, 3…).
MemoryNumber:	Specify memory number(1-128).
BMPFlags:	Specify color mode. See “3.5 BMP, CSV, Data Flags”
FileName:	Pointer to buffer containing Unicode bmpfile name.

For detailed information about memory number refer to “3.7 Display table setting”.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Note

Since memory number displayed with SLM_Ctrl_WriteDS function cannot be overwritten, you need to change displayed memory number or change displayed content with SLM_Ctrl_WriteGS function to write.

Example

```
// bmp file to memory number 1
if (SLM_Ctrl_WriteMI_BMP_A(1,1, 0, "C:¥¥test.bmp”) == SLM_OK) {
        // OK
}
```

3.2.13 SLM_Ctrl_WriteMI_CSV

```
SLM_STATUS
SLM_Ctrl_WriteMI_CSV(
   DWORD SLMNumber,
   DWORD MemoryNumber,
   DWORD CSVFlags,
   LPCWSTR FileName
)
```

Summary

Transfer CSV file(Unicode) to SLM memory.

Parameters

DisplayNumber:	Specify display number (1, 2, 3…).
MemoryNumber:	Specify memory number(1-128).
CSVFlags:	This parameter is for future option (default value 0).
FileName:	Pointer to buffer containing Unicode csvfile name. (Refer to “CSV Format3.6 CSV Format”)

For detailed information about memory number refer to “3.7 Display table setting”.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Note

Since memory number displayed with SLM_Ctrl_WriteDS function cannot be overwritten, you need to change displayed memory number or change displayed content with SLM_Ctrl_WriteGS function to write.

Example

```
// csv file to memory number 1
if (SLM_Ctrl_WriteMI_CSV(1,1, 0, _T("C:¥¥test.csv”)) == SLM_OK) {
        // OK
}
```

3.2.14 SLM_Ctrl_WriteMI_CSV_A

```
SLM_STATUS
SLM_Ctrl_WriteMI_CSV_A(
   DWORD SLMNumber,
   DWORD MemoryNumber,
   DWORD CSVFlags,
   LPCSTR FileName
)
```

Summary

Transfer CSV file(ANSI code) to SLM memory.

Parameters

DisplayNumber:	Specify display number (1, 2, 3…).
MemoryNumber:	Specify memory number (1-128).
CSVFlags:	This parameter is for future option (default value 0).
FileName:	Pointer to buffer containing ANSI code csvfile name. (Refer to “CSV Format3.6 CSV Format”)

For detailed information about memory number refer to “3.7 Display table setting”.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Note

Since memory number displayed with SLM_Ctrl_WriteDS function cannot be overwritten, you need to change displayed memory number or change displayed content with SLM_Ctrl_WriteGS function to write.

Example

```
// csv file to memory number 1
if (SLM_Ctrl_WriteMI_CSV_A(1,1, 0, "C:¥¥test.csv”) == SLM_OK) {
        // OK
}
```

3.2.15 SLM_Ctrl_WriteME

```
SLM_STATUS
SLM_Ctrl_WriteME(
   DWORD SLMNumber,
   DWORD MemoryNumber
)
```

Summary

Invalidates phase pattern stored in internal memory.

Parameters

SLMNumber:	Specify SLM number (1-8).
MemoryNumber:	Specify memory number (1-128).

For detailed information about memory number refer to “3.7 Display table setting”.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// Invalidates phase pattern stored in internal memory.
if(SLM_Ctrl_WriteME (1,1) == SLM_OK){
        // OK
}
```

3.2.16 SLM_Ctrl_WriteMT

```
SLM_STATUS
SLM_Ctrl_WriteMT(
   DWORD SLMNumber,
   DWORD TableNumber,
   DWORD MemoryNumber
)
```

Summary

Replace memory number set in display table.

Parameters

SLMNumber:	Specify SLM number (1-8).
TableNumber:	Specify table number (1-128).
MemoryNumber:	Specify memory number (1-128).

For detailed information about table number and memory number refer to “3.7 Display table setting”.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// replace memory number
if(SLM_Ctrl_WriteMT (1,1,2) == SLM_OK){
        // OK
}
```

3.2.17 SLM_Ctrl_ReadMS

```
SLM_STATUS
SLM_Ctrl_ReadMS(
   DWORD SLMNumber,
   DWORD TableNumber,
   DWORD *MemoryNumber
)
```

Summary

Read memory number set in display table.

Parameters

SLMNumber:	Specify SLM number (1, 2, 3…).
TableNumber:	Specify table number (1-128).
MemoryNumber:	Pointer to unsigned int(32bit) to store memory number value.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// Read memory mode
DWORD memorynumber;
if(SLM_Ctrl_ReadMS(1,1,& memorynumber) == SLM_OK){
        // OK
}
else{
        // Error
}
```

3.2.18 SLM_Ctrl_WriteMR

```
SLM_STATUS
SLM_Ctrl_WriteMR(
   DWORD SLMNumber,
   DWORD TableNumber1,
   DWORD TableNumber2
)
```

Summary

Write effective range of display table.

Parameters

SLMNumber:	Specify SLM number (1-8).
TableNumber1:	Specify start table number (1-128).
TableNumber2:	Specify end table number (1-128).

For detailed information about table number refer to “3.7 Display table setting”.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// effective range
if(SLM_Ctrl_WriteMR (1,1,128) == SLM_OK){
        // OK
}
```

3.2.19 SLM_Ctrl_ReadMR

```
SLM_STATUS
SLM_Ctrl_ReadMR(
   DWORD SLMNumber,
   DWORD *TableNumber1,
   DWORD *TableNumber2
)
```

Summary

Read effective range of display table.

Parameters

SLMNumber:	Specify SLM number (1-8).
TableNumber1:	Pointer to unsigned int(32bit) to store table number value.
TableNumber2:	Pointer to unsigned int(32bit) to store table number value.

For detailed information about table number refer to “3.7 Display table setting”.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// effective range
DWORD st, ed;
if(SLM_Ctrl_ReadMR (1,&st,&ed) == SLM_OK){
        // OK
}
```

3.2.20 SLM_Ctrl_WriteMP

```
SLM_STATUS
SLM_Ctrl_WriteMP(
   DWORD SLMNumber,
   DWORD TableNumber
)
```

Summary

Write table number of display table to be displayed first.

Parameters

SLMNumber:	Specify SLM number (1-8).
TableNumber:	Specify table number (1-128).

For detailed information about table number refer to “3.7 Display table setting”.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// first display table
if(SLM_Ctrl_WriteMP (1,1) == SLM_OK){
        // OK
}
```

3.2.21 SLM_Ctrl_WriteMZ

```
SLM_STATUS
SLM_Ctrl_WriteMZ(
   DWORD SLMNumber
)
```

Summary

Set contents of display table to default settings.

Parameters

SLMNumber: Specify SLM number (1-8).

For detailed information about table number refer to “3.7 Display table setting”.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// table default setting
if(SLM_Ctrl_WriteMZ (1) == SLM_OK){
        // OK
}
```

3.2.22 SLM_Ctrl_WriteMW

```
SLM_STATUS
SLM_Ctrl_WriteMW(
   DWORD SLMNumber,
   DWORD frames
)
```

Summary

Write interval for switching pattern display by number of frames.
Setting is specified by number of frames.

Parameters

SLMNumber:	Specify SLM number (1-8).
frames:	Specify frames value (0-120). e.g. 16.7ms per frame if SLM frame rate is 60Hz.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// 1s interval
if(SLM_Ctrl_WriteMW (1,60) == SLM_OK){
        // OK
}
```

3.2.23 SLM_Ctrl_ReadMW

```
SLM_STATUS
SLM_Ctrl_ReadMW(
   DWORD SLMNumber,
   DWORD *frames
)
```

Summary

Read interval for switching pattern display by number of frames.

Parameters

SLMNumber: Specify SLM number (1, 2, 3…).

frames: Pointer to unsigned int(32bit) to store frames value.(0-120)

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// frames
DWORD frames;
if(SLM_Ctrl_ReadMW(1,&frames) == SLM_OK){
        // OK
}
else{
        // Error
}
```

3.2.24 SLM_Ctrl_WriteDS

```
SLM_STATUS
SLM_Ctrl_WriteDS(
   DWORD SLMNumber,
   DWORD MemoryNumber
)
```

Summary

Specify memory number to display internal memory phase pattern.

Parameters

SLMNumber:	Specify SLM number (1, 2, 3…).
MemoryNumber:	Specify memory number (1-128).

For detailed information about memory number refer to “3.7 Display table setting”.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// display memory number1
if(SLM_Ctrl_WriteDS(1,1) == SLM_OK){
        // OK
}
else{
        // Error
}
```

3.2.25 SLM_Ctrl_ReadDS

```
SLM_STATUS
SLM_Ctrl_ReadDS(
   DWORD SLMNumber,
   DWORD *MemoryNumber
)
```

Summary

Read displayed memory number.

Parameters

SLMNumber:	Specify SLM number (1, 2, 3…).
MemoryNumber:	Pointer to unsigned int(32bit) to store memory number value.

For detailed information about memory number refer to “3.7 Display table setting”.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// Read display memory number
DWORD MemoryNumber;
if(SLM_Ctrl_ReadDS(1,&MemoryNumber) == SLM_OK){
        // OK
}
else{
        // Error
}
```

3.2.26 SLM_Ctrl_WriteDR

```
SLM_STATUS
SLM_Ctrl_WriteDR(
   DWORD SLMNumber,
   DWORD order
)
```

Summary

Display phase patterns stored in internal memory in order of display table.
Display order, range, and start position follow display table settings.
Continuous display continues until stopped by SLM_Ctrl_WriteDB function.
Some communication commands are invalid during continuous display.

Parameters

SLMNumber:	Specify SLM number (1, 2, 3…).
order:	Specify order value (0-1). 0: Descending order, 1: Ascending order

For detailed information about memory number refer to “3.7 Display table setting”.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// continuous display
if(SLM_Ctrl_WriteDR(1,1) == SLM_OK){
        // OK
}
else{
        // Error
}
Sleep(1000);

// stop
if(SLM_Ctrl_WriteDB(1) == SLM_OK){
        // OK
}
else{
        // Error
}
```

3.2.27 SLM_Ctrl_WriteDB

```
SLM_STATUS
SLM_Ctrl_WriteDB(
   DWORD SLMNumber
)
```

Summary

Stop continuous display by SLM_Ctrl_WriteDR function.

Parameters

SLMNumber: Specify SLM number (1, 2, 3…).

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// continuous display
if(SLM_Ctrl_WriteDR(1,1) == SLM_OK){
        // OK
}
else{
        // Error
}
Sleep(1000);

// stop
if(SLM_Ctrl_WriteDB(1) == SLM_OK){
        // OK
}
else{
        // Error
}
```

3.2.28 SLM_Ctrl_WriteTI

```
SLM_STATUS
SLM_Ctrl_WriteTI(
   DWORD SLMNumber,
   DWORD onoff
)
```

Summary

Write ON / OFF of trigger input value.

Parameters

SLMNumber: Specify SLM number (1-8).

onoff : Specify onoff value(0:off,1:on).

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// trigger input on
if(SLM_Ctrl_WriteTI (1,0) == SLM_OK){
        // OK
}
```

3.2.29 SLM_Ctrl_ReadTI

```
SLM_STATUS
SLM_Ctrl_ReadTI(
   DWORD SLMNumber,
   DWORD *onoff
)
```

Summary

Read ON / OFF of trigger input value.

Parameters

SLMNumber: Specify SLM number (1, 2, 3…).

onoff: Pointer to unsigned int(32bit) to store mode value(0:off,1:on).

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// read trigger input
DWORD onoff;
if(SLM_Ctrl_ReadTI(1,&onoff) == SLM_OK){
        // OK
}
else{
        // Error
}
```

3.2.30 SLM_Ctrl_WriteTM

```
SLM_STATUS
SLM_Ctrl_WriteTM(
   DWORD SLMNumber,
   DWORD onoff
)
```

Summary

Write ON / OFF of trigger output value.

Parameters

SLMNumber: Specify SLM number (1-8).

onoff : Specify onoff value(0:off,1:on).

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// trigger output on
if(SLM_Ctrl_WriteTM (1,1) == SLM_OK){
        // OK
}
```

3.2.31 SLM_Ctrl_ReadTM

```
SLM_STATUS
SLM_Ctrl_ReadTM(
   DWORD SLMNumber,
   DWORD *onoff
)
```

Summary

Read ON / OFF of trigger output value.

Parameters

SLMNumber: Specify SLM number (1, 2, 3…).

onoff: Pointer to unsigned int(32bit) to store mode value(0:off,1:on).

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// read trigger output
DWORD onoff;
if(SLM_Ctrl_ReadTM(1,&onoff) == SLM_OK){
        // OK
}
else{
        // Error
}
```

3.2.32 SLM_Ctrl_WriteTC

```
SLM_STATUS
SLM_Ctrl_WriteTC(
   DWORD SLMNumber,
   DWORD order
)
```

Summary

Write ascending / descending order of pattern display by trigger input.

Parameters

SLMNumber: Specify SLM number (1-8).

order : Specify order value(0-1).

0: Descending order, 1: Ascending order

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// trigger display order
if(SLM_Ctrl_WriteTC (1,0) == SLM_OK){
        // OK
}
```

3.2.33 SLM_Ctrl_ReadTC

```
SLM_STATUS
SLM_Ctrl_ReadTC(
   DWORD SLMNumber,
   DWORD *order
)
```

Summary

Read ascending / descending order of pattern display by trigger input.

Parameters

SLMNumber:	Specify SLM number (1, 2, 3…).
order:	Pointer to unsigned int(32bit) to store order value. 0: Descending order, 1: Ascending order

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.
(Refer to “3.4 SLM_STATUS”)

Example

```
// Read trigger order
DWORD order;
if(SLM_Ctrl_ReadTC(1,&order) == SLM_OK){
        // OK
}
else{
        // Error
}
```

3.2.34 SLM_Ctrl_WriteTS

```
SLM_STATUS
SLM_Ctrl_WriteTS(
   DWORD SLMNumber
)
```

Summary

Performs same operation as trigger input.

Parameters

SLMNumber: Specify SLM number (1-8).

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// software trigger
if(SLM_Ctrl_WriteTS (1) == SLM_OK){
        // OK
}
```

3.2.35 SLM_Ctrl_ReadT

```
SLM_STATUS
SLM_Ctrl_ReadT(
   DWORD SLMNumber,
   INT32 *driveboardTemp,
   INT32 *optionboardTemp
)
```

Summary

Read drive board and option board Celsius temperatures.

Parameters

SLMNumber:	Specify SLM number (1, 2, 3…).
driveboardTemp:	Pointer to int(32bit) to store driveboardTemp value multiplied by 10.
optionboardTemp:	Pointer to int(32bit) to store optionboardTemp value multiplied by 10.

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// Read display mode
int dTemp,oTemp;
if(SLM_Ctrl_ReadT(1,&dTemp,&oTemp) == SLM_OK){
        // OK
        printf(“Drive Board %0.1f degrees, Option Board %0.1f degrees”,
                ((float) dTemp)/10, ((float) oTemp)/10);
}
else{
        // Error
}
```

3.2.36 SLM_Ctrl_ReadEDO

```
SLM_STATUS
SLM_Ctrl_ReadEDO(
   DWORD SLMNumber,
   DWORD *driveboardError,
   DWORD *optionboardError
)
```

Summary

Read error flags of "Drive board" and "Option board".

These error values are output in hexadecimal.

Parameters

SLMNumber:	Specify SLM number (1, 2, 3…).
driveboardError:	Pointer to unsigned int(32bit) to store driveboard error value.
optionboardError:	Pointer to unsigned int(32bit) to store optionboard error value.

7	6	5	4	3	2	1	0
0	0	0	0	(4)	(3)	(2)	(1)

(1) Startup error 1 (Drive board)
(2) Startup error 2 (Drive board)
(3) Video signal error (No signal)
(4) Temperature error (70°C or higher)

<example>
Normal 0000h
Error 0008h (Temperature error)

Fig. 3.2-1 drive board error

7	6	5	4	3	2	1	0
0	0	0	0	(4)	(3)	(2)	(1)

(1) Startup error 1 (Option board)
(2) Startup error 2 (Option board)
(3) Voltage level error (DC 5.0V)
(4) Temperature error (70°C or higher)

<example>
Normal 0000h
Error 0004h (Voltage level error)

Fig. 3.2-2 option board error

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// Read error
DWORD driveerr, optionerr;
if(SLM_Ctrl_ReadEDO(1,&driveerr,&optionerr) == SLM_OK){
        // OK
        printf("Drive Board Error %4X, Option Board Error %4X¥n",driveerr, optionerr);
}
else{
        // Error
}
```

3.2.37 SLM_Ctrl_ReadSDO

```
SLM_STATUS
SLM_Ctrl_ReadSDO(
   DWORD SLMNumber,
   LPSTR driveboardID,
   LPSTR optionboardID
)
```

Summary

Read identification numbers of "Drive board" and "Option board".

Parameters

SLMNumber:	Specify SLM number (1, 2, 3…).
driveboardID:	Pointer to a 16-byte buffer to store driveboardID
optionboardID:	Pointer to a 16-byte buffer to store optionboardID

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// Read display mode
char driveboardID[16];
char optionboardID[16];
if(SLM_Ctrl_ReadSDO(1,driveboardID, optionboardID) == SLM_OK){
        // OK
        // driveboardID => “18050004”
        // optionboardID => “18050004”
}
else{
        // Error
}
```

3.3 Finalizing

3.3.1 SLM_Ctrl_Close

```
SLM_STATUS
SLM_Ctrl_Close (
   DWORD SLMNumber,
)
```

Summary

Close USB interface.

Parameters

SLMNumber: Specify SLM number (1-8).

Return Value

SLM_OK if successful, otherwise SLM_STATUS error code is returned.

(Refer to “3.4 SLM_STATUS”)

Example

```
// Open USB interface
if(SLM_Ctrl_Close (1) == SLM_OK){
        // OK
}
```

3.4 SLM_STATUS

SLM_STATUS is obtained as a return value when SLM function is executed.
You can use this return value to check SLM status.

Table 3.4-1 SLM_STATUS

Defined name	Return Value	Note
SLM_OK	0	
SLM_NG	1	NG
SLM_BS	2	SLM is Busy
SLM_ER	3	parameter error
SLM_INVAID_MONITOR	-1	not find display no
SLM_NOT_OPEN_MONITOR	-2	not open display
SLM_OPEN_WINDOW_ERR	-3	window open error
SLM_DATA_FORMAT_ERR	-4	data format error
SLM_FILE_READ_ERR	-101	over 1023
SLM_NOT_OPEN_USB	-200	not open USB
SLM_OTHER_ERROR	-1000	other error
FT_INVALID_HANDLE	-10001	USB driver error.
FT_DEVICE_NOT_FOUND	-10002	Check connected device's power. If connected, reset the power.
FT_DEVICE_NOT_OPENED	-10003	Already opened.
FT_IO_ERROR	-10004	USB driver error.
FT_INSUFFICIENT_RESOURCES	-10005	USB driver error.
FT_INVALID_PARAMETER	-10006	USB driver error.
FT_INVALID_BAUD_RATE	-10007	USB driver error.
FT_DEVICE_NOT_OPENED_FOR_ERASE	-10008	USB driver error.
FT_DEVICE_NOT_OPENED_FOR_WRITE	-10009	USB driver error.
FT_FAILED_TO_WRITE_DEVICE	-10010	USB driver error.
FT_EEPROM_READ_FAILED	-10011	USB driver error.
FT_EEPROM_WRITE_FAILED	-10012	USB driver error.
FT_EEPROM_ERASE_FAILED	-10013	USB driver error.
FT_EEPROM_NOT_PRESENT	-10014	USB driver error.
FT_EEPROM_NOT_PROGRAMMED	-10015	USB driver error.
FT_INVALID_ARGS	-10016	USB driver error.
FT_NOT_SUPPORTED	-10017	USB driver error.
FT_NO_MORE_ITEMS	-10018	USB driver error.
FT_TIMEOUT	-10019	USB driver error.
FT_OPERATION_ABORTED	-10020	USB driver error.
FT_RESERVED_PIPE	-10021	USB driver error.
FT_INVALID_CONTROL_REQUEST_DIRECTION	-10022	USB driver error.
FT_INVALID_CONTROL_REQUEST_TYPE	-10023	USB driver error.
FT_IO_PENDING	-10024	USB driver error.
FT_IO_INCOMPLETE	-10025	USB driver error.
FT_HANDLE_EOF	-10026	USB driver error.
FT_BUSY	-10027	USB driver error.
FT_NO_SYSTEM_RESOURCES	-10028	USB driver error.
FT_DEVICE_LIST_NOT_READY	-10029	USB driver error.
FT_DEVICE_NOT_CONNECTED	-10030	USB driver error.
FT_INCORRECT_DEVICE_PATH	-10031	USB driver error.
FT_OTHER_ERROR	-10032	USB driver error.

3.5 BMP, CSV, Data Flags

Table 3.5-1 : BMP Flags

Status Name	Flags Value 8bit color	Flags Value 10bit color (10bit is simply 4 times 8bit)	define name	Note
Original color display	—	0x0	FLAGS_COLOR_NOP	BMP only.
Only red color display	0x01	0x101	FLAGS_COLOR_R	BMP only.
Only green color display	0x02	0x102	FLAGS_COLOR_G	BMP only.
Only blue color display	0x04	0x104	FLAGS_COLOR_B	BMP only.
Convert grayscale (Y=0.299R+0.587G+0.114B)	0x08	0x108	FLAGS_COLOR_GRAY	BMP only.
Rate 120Hz SLM	0x20000000	0x20000100	FLAGS_RATE120	

BMP Data Format

bit	7	6	5	4	3	2	1	0
Red	R7	R6	R5	R4	R3	R2	R1	R0
Green	G7	G6	G5	G4	G3	G2	G1	G0
Bule	B7	B6	B5	B4	B3	B2	B1	B0

Original color display (Use Red 3bit,Green 3bit,Bule 4bit)

bit	9	8	7	6	5	6	3	2	1	0
LCOS Format	R7	R6	R5	G7	G6	G5	B7	B6	B5	B4

Only red color display

bit	9	8	7	6	5	6	3	2	1	0
LCOS Format	0	0	R7	R6	R5	R4	R3	R2	R1	R0

Only red color display & 10bit color

bit	9	8	7	6	5	6	3	2	1	0
LCOS Format	R7	R6	R5	R4	R3	R2	R1	R0	0	0

Only green color display

bit	9	8	7	6	5	6	3	2	1	0
LCOS Format	0	0	G7	G6	G5	G4	G3	G2	G1	G0

Only green color display & 10bit color

bit	9	8	7	6	5	6	3	2	1	0
LCOS Format	G7	G6	G5	G4	G3	G2	G1	G0	0	0

Only blue color display

bit	9	8	7	6	5	6	3	2	1	0
LCOS Format	0	0	B7	B6	B5	B4	B3	B2	B1	B0

Only blue color display & 10bit color

bit	9	8	7	6	5	6	3	2	1	0
LCOS Format	B7	B6	B5	B4	B3	B2	B1	B0	0	0

Convert grayscale

bit	9	8	7	6	5	6	3	2	1	0
LCOS Format	0	0	G7	G6	G5	G4	G3	G2	G1	G0

Convert grayscale & 10bit color

bit	9	8	7	6	5	6	3	2	1	0
LCOS Format	G7	G6	G5	G4	G3	G2	G1	G0	0	0

120Hz model

FLAGS_RATE120 bit OFF.

First frame

bit	9	8	7	6	5	6	3	2	1	0
LCOS Format	R7	R6	R5	G7	G6	G5	B7	B6	B5	B4

Next frame

bit	9	8	7	6	5	6	3	2	1	0
LCOS Format	R3	R2	R1	G3	G2	G1	B3	B2	B1	B0

FLAGS_RATE120 bit ON.

First frame

bit	9	8	7	6	5	6	3	2	1	0
LCOS Format	R7	R6	R5	G7	G6	G5	B7	B6	B5	B4

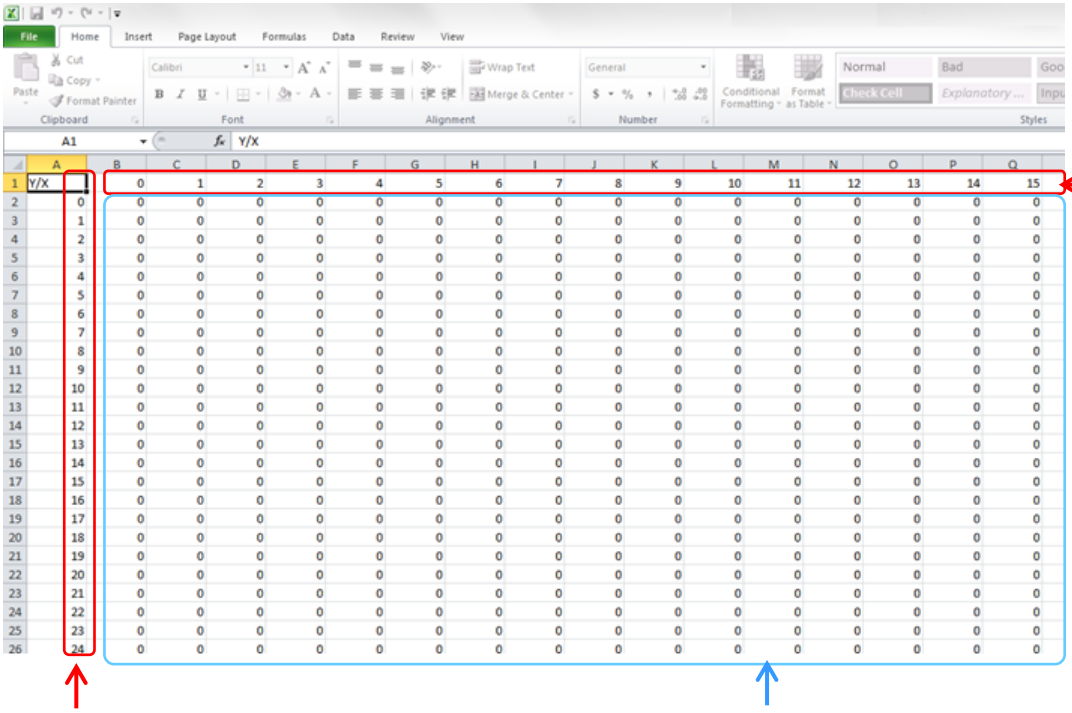
Next frame

bit	9	8	7	6	5	6	3	2	1	0
LCOS Format	R7	R6	R5	G7	G6	G5	B7	B6	B5	B4

Fig. 3.5-1 10bit encoding format in RGB color

3.6 CSV Format

The CSV file with data format made in accordance with Fig. 3.6-1 can be opened.
The data can be edited using spreadsheet like Microsoft excel.

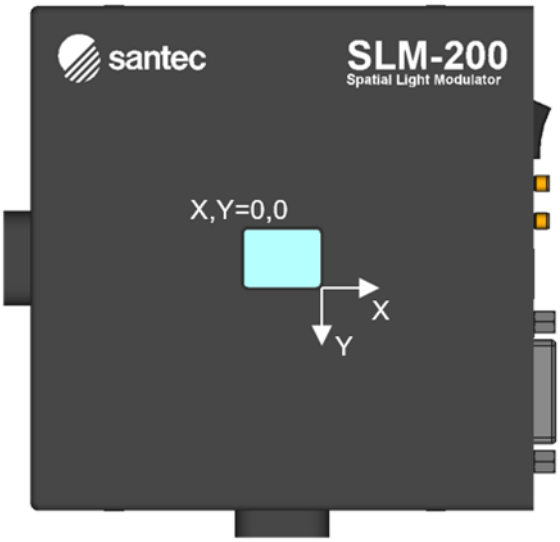


Y/X	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

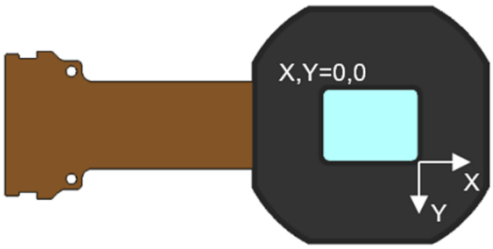
X: Horizontal pixel number

Y: Vertical pixel number

Gray scale level for each pixel: 0~1023 (10 bit)
0 to 1023 corresponds to 0 to 2pi at specified wave length.



(a) All-in-one model



(b) Separate model (LCOS unit)

Fig. 3.6-1 : Data format of pattern files.

3.7 Display table setting

<Default>

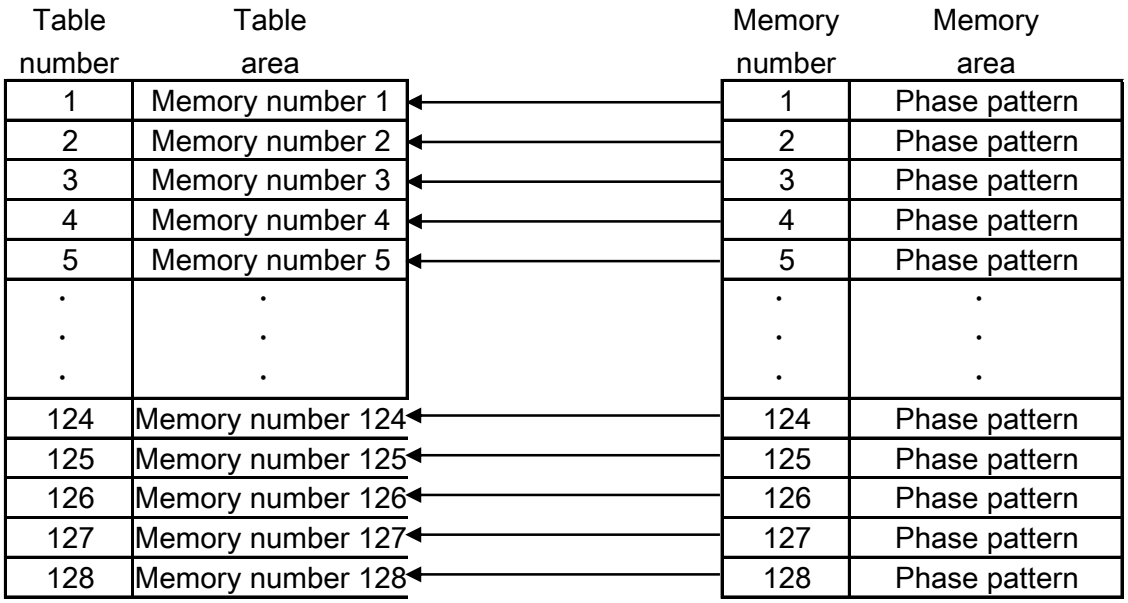


Fig. 3.7-1 : Default table

<Display table change>

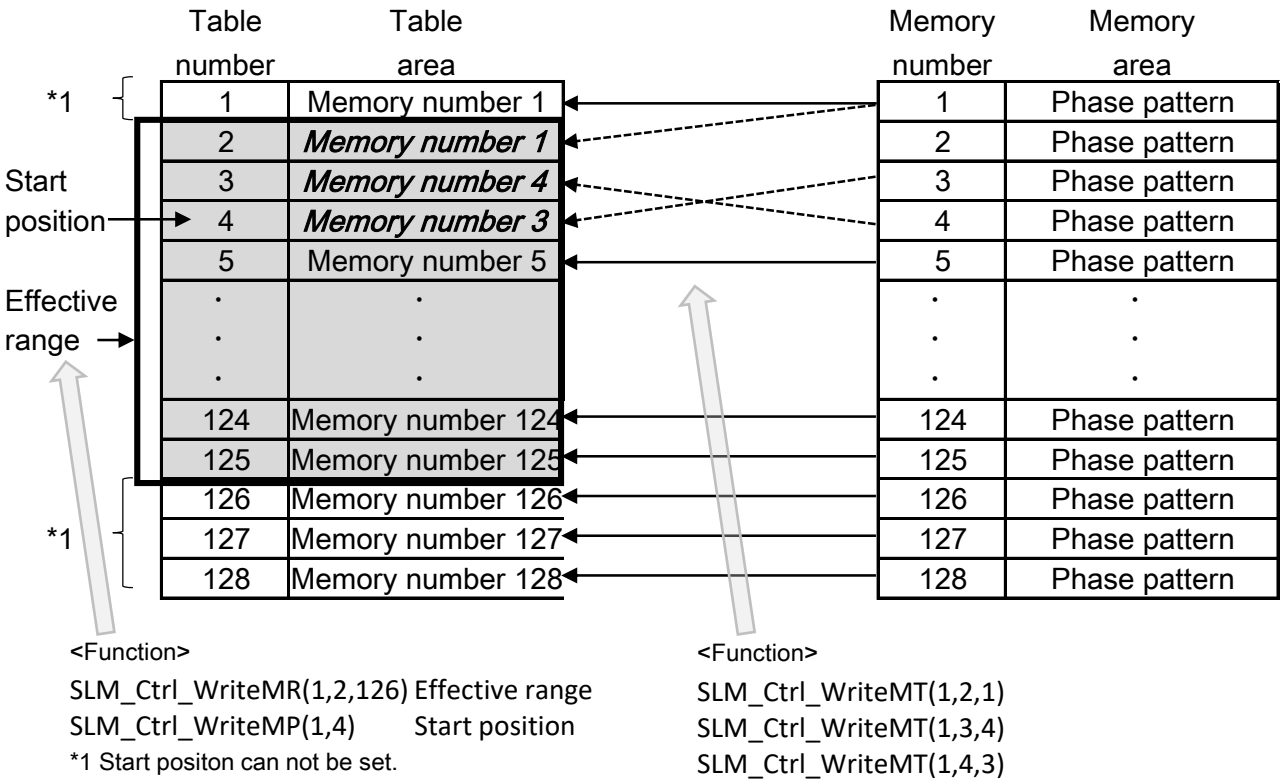


Fig. 3.7-2 : Display table changed

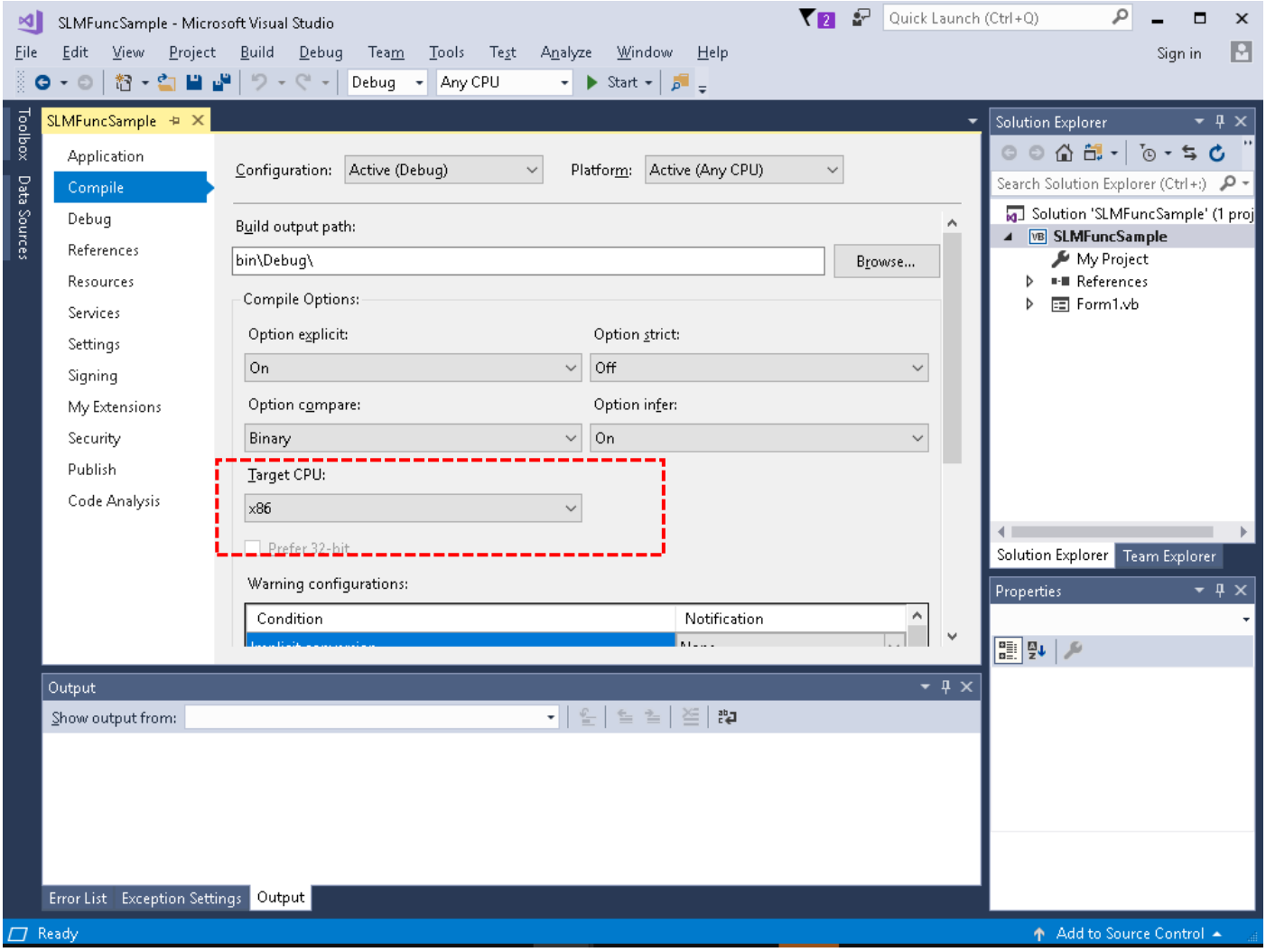
4 Samples

4.1 VB.net

4.1.1 Project Setting

32bit: “Compile -> Target CPU” to x86.

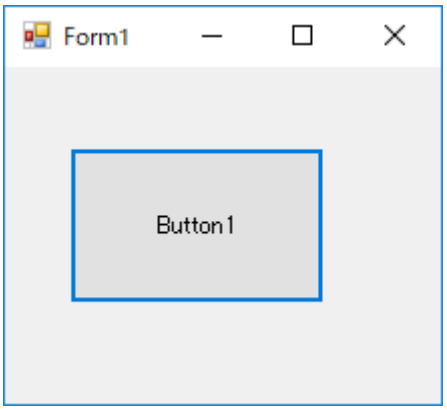
64bit: “Compile -> Target CPU” to x64.



4.1.2 Sample source

```
Imports System.Runtime.InteropServices

Module SLMFunc

  '*****************************************************************************
    '/ SLM Status Codes
    '*****************************************************************************/
    Public Enum SLM_STATUS As Integer
        SLM_OK = 0                          ' OK
        SLM_INVAID_MONITOR = -1         ' not find display no
        SLM_NO_OPEN_MONITOR = -2        ' not open display
        SLM_OPEN_WINDOW_ERR = -3        ' window open error
        SLM_DATA_FORMAT_ERR = -4        ' data format error

        SLM_FILE_READ_ERR = -101        ' not find file
        SLM_OTHER_ERROR = -1000         ' other Error
    End Enum

    Private Const DLLFileName As String = "SLMFunc.dll"

    <System.Runtime.InteropServices.DllImport(DLLFileName,
CallingConvention:=Runtime.InteropServices.CallingConvention.Cdecl)>
    Function SLM_Disp_Info(ByVal DisplayNumber As UInt32, ByRef width As UShort, ByRef height As UShort) As Int32
    End Function

    <System.Runtime.InteropServices.DllImport(DLLFileName,
CallingConvention:=Runtime.InteropServices.CallingConvention.Cdecl)>
    Function SLM_Disp_Open(ByVal DisplayNumber As UInt32) As Int32
    End Function

    <System.Runtime.InteropServices.DllImport(DLLFileName,
CallingConvention:=Runtime.InteropServices.CallingConvention.Cdecl)>
    Function SLM_Disp_Close(ByVal DisplayNumber As UInt32) As Int32
    End Function


    <System.Runtime.InteropServices.DllImport(DLLFileName,
CallingConvention:=Runtime.InteropServices.CallingConvention.Cdecl)>
    Function SLM_Disp_GrayScale(ByVal DisplayNumber As UInt32, ByVal Flags As UInt32, ByVal GrayScale As UShort)
As Int32
    End Function

    <System.Runtime.InteropServices.DllImport(DLLFileName,
CallingConvention:=Runtime.InteropServices.CallingConvention.Cdecl)>
    Function SLM_Disp_BMP(ByVal DisplayNumber As UInt32, ByVal Flags As UInt32, ByVal b As IntPtr) As Int32
    End Function

    <System.Runtime.InteropServices.DllImport(DLLFileName,
CallingConvention:=Runtime.InteropServices.CallingConvention.Cdecl)>
    Function SLM_Disp_Data(ByVal DisplayNumber As UInt32, ByVal width As UInt32, ByVal height As UInt16, ByVal Flags
As UInt32, ByVal data() As UShort) As Int32
    End Function


    <System.Runtime.InteropServices.DllImport(DLLFileName,
CallingConvention:=Runtime.InteropServices.CallingConvention.Cdecl)>
    Function SLM_Disp_ReadBMP(ByVal DisplayNumber As Integer, ByVal Flags As UInt32,
<MarshalAs(UnmanagedType.LPWStr)> ByVal Para1 As String) As Integer
    End Function

    <System.Runtime.InteropServices.DllImport(DLLFileName,
CallingConvention:=Runtime.InteropServices.CallingConvention.Cdecl)>
    Function SLM_Disp_ReadCSV(ByVal DisplayNumber As Integer, ByVal Flags As UInt32,
<MarshalAs(UnmanagedType.LPWStr)> ByVal Para1 As String) As Integer

End Function
```

```
Public Class Form1

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

        If (SLM_Disp_Open(2) = SLM_STATUS.SLM_OK) Then

            SLM_Disp_GrayScale(2, 0, 100)
            System.Threading.Thread.Sleep(1000)
            SLM_Disp_Close(2)

        End If

    End Sub
End Class
```



Displayed all pixel in grayscale 100 when click “Button 1”.

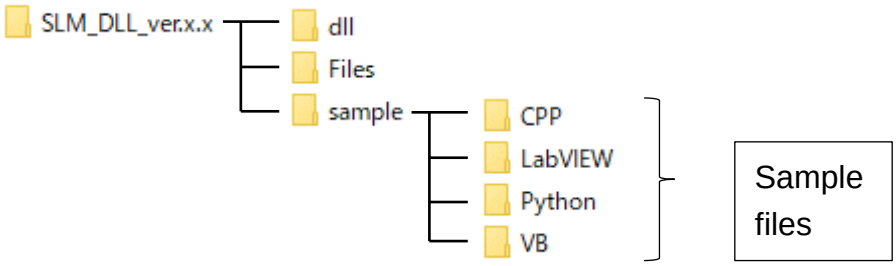
4.2 Python 3.6 Sample source

```
# ==============================================================================
# SLMFunc.dll Sample Program for Python
# ==============================================================================
# coding:utf-8
import ctypes
import time
import numpy as np

# ==============================================================================
# 2D gradation
# ==============================================================================
def get_gradation_2d(start, stop, width, height, is_horizontal):
    if is_horizontal:
        return np.tile(np.linspace(start, stop, width), (height, 1))
    else:
        return np.tile(np.linspace(start, stop, height), (width, 1)).T

# ==============================================================================
# Main
# ==============================================================================
dNo = 1
sampleFolder = 'C:¥¥workspace¥¥SLM¥¥Files¥¥2k¥¥'

# ==============================================================================
# set dll
# ==============================================================================
dll = ctypes.cdll.LoadLibrary('SLMFunc.dll')

# ==============================================================================
# open
# SLM_STATUS SLM_Disp_Open(DWORD dNo)
# ==============================================================================
dll.SLM_Disp_Open(ctypes.c_int32(dNo))

# ==============================================================================
# grayscale test
# SLM_STATUS SLM_Disp_GrayScale(DWORD dNo, DWORD type, USHORT GrayScale)
# ==============================================================================
for i in range(10):
    ret = dll.SLM_Disp_GrayScale(ctypes.c_int32(dNo),ctypes.c_int32(0), ctypes.c_int(i*10))
    if(ret != 0): print(ret)
    time.sleep(0.05)
time.sleep(0.1)

# ==============================================================================
# array data test
# SLM_STATUS SLM_Disp_Data(DWORD dNo, USHORT width, USHORT height, DWORD type, short* data)
# ==============================================================================
n = get_gradation_2d(0,1023,1920,1200,1)
n1 = n.astype(np.int16)
n_h, n_w = n1.shape    # height, width

for i in range(5):
    n1 = np.roll(n1,10)
    c = n1.ctypes.data_as(ctypes.POINTER((ctypes.c_int16 * n_h) * n_w)).contents    # convert
    ret = dll.SLM_Disp_Data(ctypes.c_int32(dNo),ctypes.c_int16(n_w),ctypes.c_int16(n_h),ctypes.c_int32(0),c)
    if(ret != 0): print(ret)
    time.sleep(0.05)

time.sleep(0.5)

# ==============================================================================
# bmp data test
# SLM_STATUS SLM_Disp_BMP(DWORD dNo, DWORD type, HBITMAP bmp)
# no sample
# ==============================================================================
#time.sleep(0.5)

# ==============================================================================
```

```
# bmp file test
# SLM_STATUS SLM_Disp_ReadBMP(DWORD dNo, DWORD type, LPWSTR FileName)
# ==============================================================================
print('BMP File')
filename=sampleFolder + 'Laguerre-Gaussian(LG0,-1).bmp'
ret = dll.SLM_Disp_ReadBMP(ctypes.c_int32(dNo),ctypes.c_int32(0),filename)
time.sleep(1)

# ==============================================================================
# csv file test
# ==============================================================================
print('CSV File')
filename = sampleFolder + 'santec_logo.csv'
ret = dll.SLM_Disp_ReadCSV(ctypes.c_int32(dNo),ctypes.c_int32(0),filename)
time.sleep(1)

# ==============================================================================
# close
# SLM_STATUS SLM_Disp_ReadCSV(DWORD dNo, DWORD type, LPWSTR FileName)
# ==============================================================================
dll.SLM_Disp_Close(ctypes.c_int(dNo))
```

4.3 Other sample source

Sample files are in the following location after extracting distribution file (SLM_DLL_ver.x.x.zip).



5 Revision History

Table 5 Revision History

Revision	Changes	Date
2.0	Initial Release	2020.04.13
2.4	Add 120Hz Option. SLM_Disp_GrayScale, SLM_Disp_Data, SLM_Disp_ReadBMP, SLM_Disp_ReadBMP_A, SLM_Disp_ReadCSV, SLM_Disp_ReadCSV_A	2021.07.13

6 Contact

WARNING!

In the event of any trouble with this product, turn the unit off in accordance with the procedures to shut off the power described in this operation manual, disconnect the power source cord, record the product name and serial number described on the name plate of the product, and then contact our dealer at your place or directly contact us at Santec Photonics Laboratories. Our telephone number and facsimile number are shown below. However, we are not responsible for any trouble arising from your own repair or modification on this product.

5823 Ohkusa-Nenjyozaka, Komaki, Aichi 485-0802, Japan

SANTEC CORPORATION

Tel. +81-568-79-1959

Fax +81-568-79-1718

433 Hackensack Ave., Hackensack, NJ 07601, U.S.A.

SANTEC U.S.A. CORPORATION

Toll Free +1-201-488-5505

Fax +1-201-488-7702

Grand Union Studios, 332 Ladbroke Grove, London W10 5AD

SANTEC EUROPE LIMITED

Tel. +44-20-3176-1550

11F Room E, Hua Du Bldg., No.838 Zhangyang Road, Pudong, Shanghai 200122 China

SANTEC (SHANGHAI) Co., Ltd

Tel. +86-21-58361261, +86-21-58361262

Fax +86-21-58361263

www.santec.com