SC10®
Benchtop Shutter Controller

SDK Manual

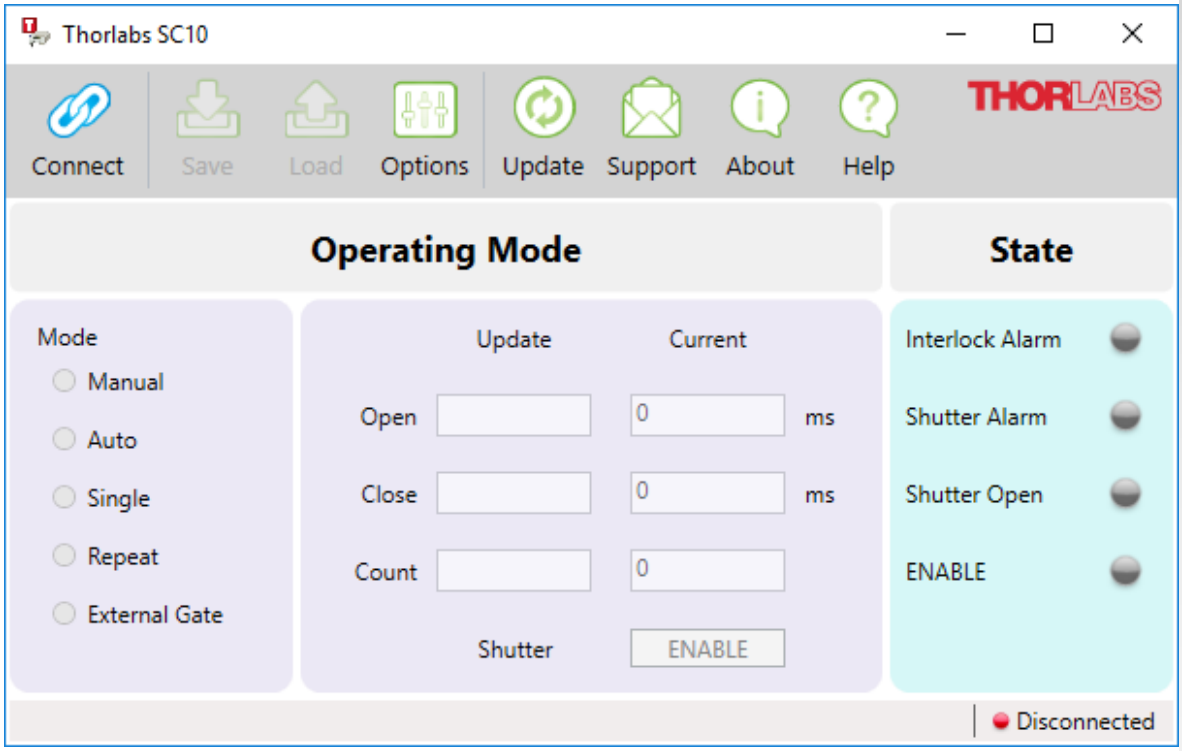
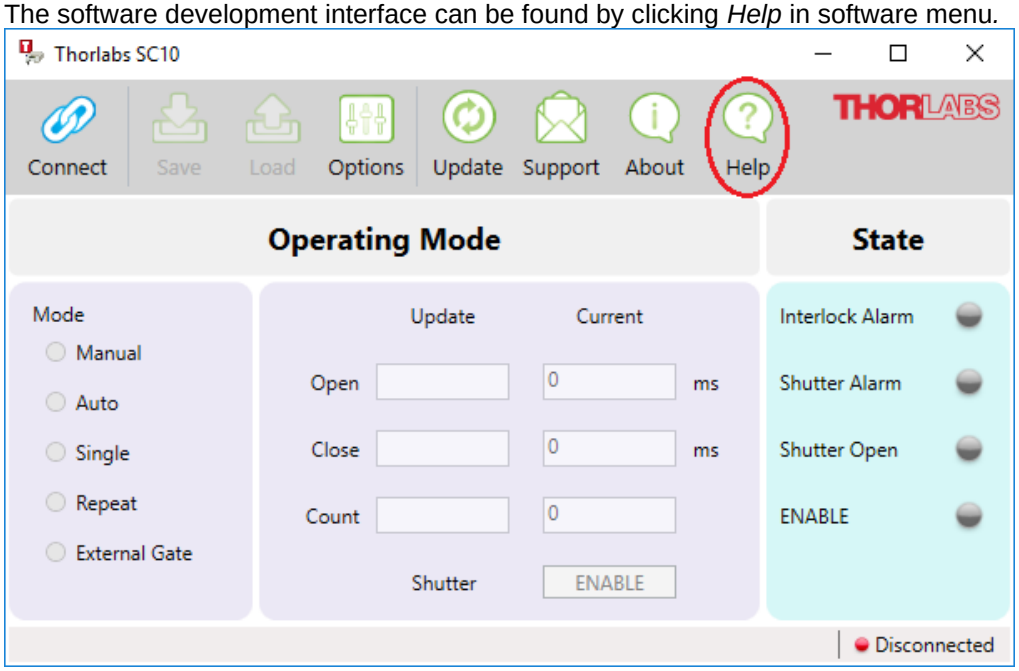


Table of Contents

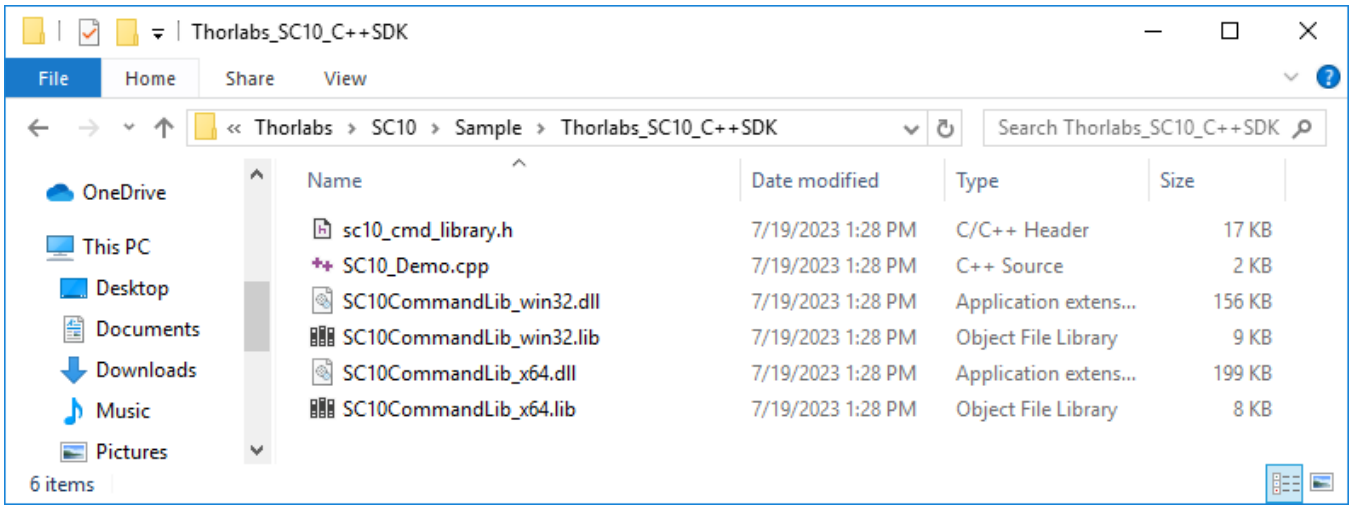
Chapter 1 Introduction

User can start software development in C/C++ develop environment, LabVIEW and Python.
The software development interface can be found by clicking *Help* in software menu.



Chapter 2 C++ Software Development Kit

The user can start software development with Visual Studio 2017 or later versions. The supported files are in **\Thorlabs_SC10_C++SDK** under the **Sample** directory.



Copy **SC10CommandLib_win32.dll** or **SC10CommandLib_x64.dll** to your program folder, and make sure the library file(sc10_cmd_library.h) is in the same folder.

Below is the description of the header file **sc10_cmd_library.h**:

2.1. sc10_cmd_library.h File Reference

2.1.1. Functions

- **COMMANDLIB_API** int **List** (unsigned char *serialNo, int size)
  list all the possible port on this computer.
- **COMMANDLIB_API** int **Open** (char *serialNo, int nBaud, int timeout)
  open port function.
- **COMMANDLIB_API** int **IsOpen** (char *serialNo)
  check opened status of port
- **COMMANDLIB_API** int **Close** (int hdl)
  close current opend port
- **COMMANDLIB_API** int **Read** (int hdl, unsigned char *b, int limit)
- **COMMANDLIB_API** int **Write** (int hdl, char *b, int size)
- **COMMANDLIB_API** int **Set** (int hdl, char *c, int size)
- **COMMANDLIB_API** int **Get** (int hdl, char *c, unsigned char *d)
- **COMMANDLIB_API** int **Purge** (int hdl, int flag)
  Purge the RX and TX buffer on port.
- **COMMANDLIB_API** int **SetBaudRate** (int hdl, int value)
- **COMMANDLIB_API** int **SetMode** (int hdl, int value)
- **COMMANDLIB_API** int **ToggleEnable** (int hdl)

- **COMMANDLIB_API** int **SetOpenTime** (int hdl, int value)
- **COMMANDLIB_API** int **SetCloseTime** (int hdl, int value)
- **COMMANDLIB_API** int **SetTriggerMode** (int hdl, int value)
- **COMMANDLIB_API** int **SetExternalTriggerMode** (int hdl, int value)
- **COMMANDLIB_API** int **SetRepeatCount** (int hdl, int value)
- **COMMANDLIB_API** int **GetBaudRate** (int hdl, int *value)
- **COMMANDLIB_API** int **GetMode** (int hdl, int *value)
- **COMMANDLIB_API** int **GetEnableState** (int hdl, int *value)
- **COMMANDLIB_API** int **GetOpenTime** (int hdl, int *value)
- **COMMANDLIB_API** int **GetCloseTime** (int hdl, int *value)
- **COMMANDLIB_API** int **GetTriggerMode** (int hdl, int *value)
- **COMMANDLIB_API** int **GetExternalTriggerMode** (int hdl, int *value)
- **COMMANDLIB_API** int **GetRepeatCount** (int hdl, int *value)
- **COMMANDLIB_API** int **GetClosedState** (int hdl, int *value)
- **COMMANDLIB_API** int **GetInterlockTripped** (int hdl, int *value)
- **COMMANDLIB_API** int **GetStat** (int hdl, char *d)
  Undocumented Command.
- **COMMANDLIB_API** int **GetId** (int hdl, char *d)
- **COMMANDLIB_API** int **GetCommands** (int hdl, char *d)
- **COMMANDLIB_API** int **SaveSettings** (int hdl)
- **COMMANDLIB_API** int **StoreConfiguration** (int hdl)
- **COMMANDLIB_API** int **LoadConfiguration** (int hdl)

2.1.2. Function Documentation

COMMANDLIB_API int Close (int *hdl*)

close current opend port

Parameters:

hdl	handle of port.

Returns:

0: success; negtive number : failed.

COMMANDLIB_API int GetBaudRate (int *hdl*, int * value)

Get baud rate

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
value	0: 9.6k, 1:115k

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int GetClosedState (int *hdl*, int * value)

Get closed state

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.

value	1: shutter is closed, 0: shutter is open

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int GetCloseTime (int *hdl*, int * value)

Get close duration

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
value	the shutter's close time in ms

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int GetEnableState (int *hdl*, int * value)

Get State

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
speed	0: the shutter is disabled, 1: enabled

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int GetExternalTriggerMode (int *hdl*, int * value)

Get Ex trigger mode

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
value	external trigger mode

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int GetId (int *hdl*, char * *d*)

get the SC10 id

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
d	output string (<255)

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int GetInterlockTripped (int *hdl*, int * value)

Get interlock tripped

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
value	1: interlock is tripped, otherwise 0

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int GetMode (int *hdl*, int * value)

Get operating mode

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
value	the mode value(1-5)

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int GetOpenTime (int *hdl*, int * value)

Get open duration

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
value	the shutter's open time in ms

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int GetPorts (char * *serialNo*)

list all the possible port on this computer.

Parameters:

serialNo	port list returned string include serial number and device descriptor, seperated by comma

Returns:

non-negtive number: number of device in the list; negtive number : failed.

COMMANDLIB_API int GetRepeatCount (int *hdl*, int * value)

Return the repeat count

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
value	repeat count, a value of 1-99

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int GetTriggerMode (int *hdl*, int * value)

Get trigger mode

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
value	0:internal trigger mode,1:external trigger mode

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int IsOpen (char * *serialNo*)

check opened status of port

Parameters:

serialNo	serial number of the device to be checked.

Returns:

0: port is not opened; 1 : port is opened.

COMMANDLIB_API int List (char * *serialNo*)

list all the possible port on this computer.

Parameters:

serialNo	port list returned string include serial number and device descriptor, separated by comma

Returns:

non-negative number: number of device in the list; negative number: failed.

COMMANDLIB_API int LoadConfiguration (int *hdl*)

Load configuration from EEPROM

make sure the port was opened SUCCESSful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.

Returns:

0: SUCCESS; negative number: failed.

COMMANDLIB_API int Open (char * *serialNo*, int *nBaud*, int *timeout*)

open port function.

Parameters:

serialNo	serial number of the device to be opened, use GetPorts function to get exist list first.
nBaud	bit per second of port
timeout	set timeout value in (s)

Returns:

non-negtive number: hdl number returned successfully; negtive number : failed.

COMMANDLIB_API int SaveSettings (int *hdl*)

Save current baud rate and output trigger mode

make sure the port was opened SUCCESSful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.

Returns:

0: SUCCESS; negative number: failed.

COMMANDLIB_API int SetBaudRate (int *hdl*, int *value*)

set SC10's serial baud rate

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
value	SC10 baud rate, 0 for 9.6k and 1 for 115k

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int SetCloseTime (int *hdl*, int *value*)

set close duration

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
value	shutter's close time in ms

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int SetExternalTriggerMode (int *hdl*, int *value*)

Set Ex Trigger mode

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
value	0:set the output trigger to follow the shutter output when the SH05 is connected, 1: force the trigger output to follow the controller output when an SH05 is equipped

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int SetMode (int *hdl*, int *value*)

Set operating mode

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
value	SC10 mode. 1-manual, 2-auto, 3-single, 4-repeat, 5-external gate

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int SetOpenTime (int *hdl*, int *value*)

set open duration

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
value	shutter's open time in ms

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int SetRepeatCount (int *hdl*, int *value*)

Set repeat count

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
value	set the repeat count when in repeat mode, a value of 1-99

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int SetTimeout (int *hdl*, int *timeout*)

set SC10's timeout

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
timeout	timeout

Returns:

0: success;

COMMANDLIB_API int SetTriggerMode (int *hdl*, int *value*)

set the trigger mode

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.
value	0:internal trigger mode,1:external trigger mode

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

COMMANDLIB_API int StoreConfiguration (int *hdl*)

Store configuration, save current settings(ex. mode, open time, close time) into EEPROM

make sure the port was opened SUCCESSful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

hdl	handle of port.

Returns:

0: SUCCESS; negative number: failed.

COMMANDLIB_API int ToggleEnable (int *hdl*)

Enable/Disable the shutter

make sure the port was opened successful before call this function.

make sure this is the correct device by checking the ID string before call this function.

Parameters:

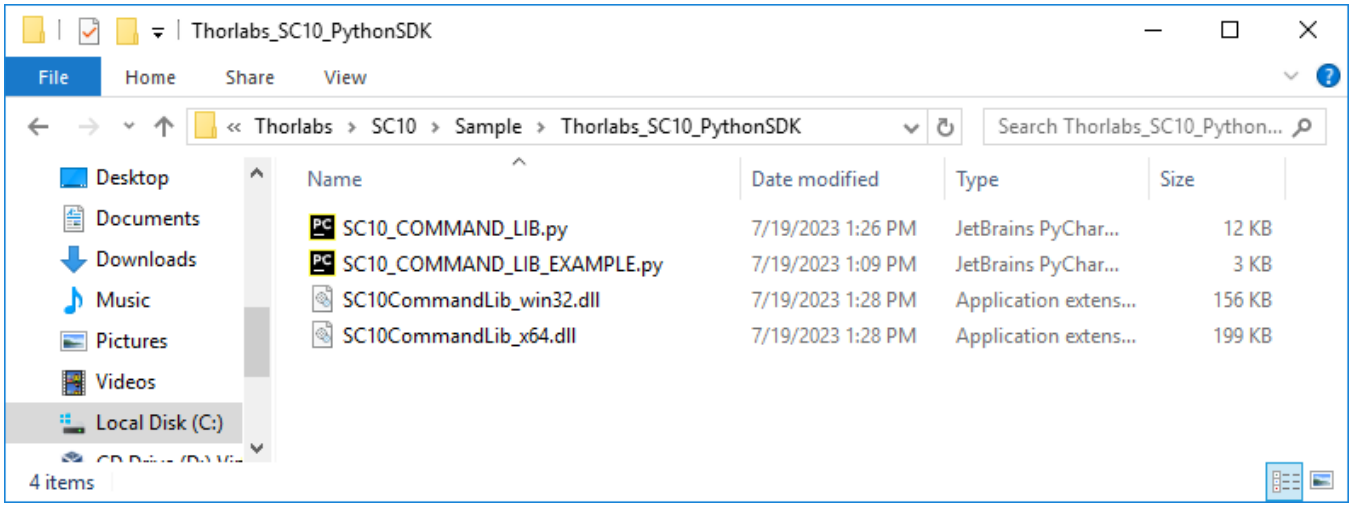
hdl	handle of port.

Returns:

0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

Chapter 3 Python Software Development Kit

The user can start software development with Python 3.7 or later versions. The supported files are in **\Thorlabs_SC10_PythonSDK** under the **Sample** directory.



Copy all the files to your python project, and make sure they are in the same folder. The **SC10_COMMAND_LIB.py** is the wrapper for **SC10_COMMAND_LIB**(**SC10CommandLib_win32.dll / SC10CommandLib_x64.dll** in C/C++ development environment). The **SC10_COMMAND_LIB_EXAMPLE.py** is the example code for how to use the python APIs.

User can import the **SC10_COMMAND_LIB** in 32-bit application with **SC10CommandLib_win32.dll** or 64-bit application with **SC10CommandLib_x32.dll.** If in 64-bit application, then you need to modify the **__init__** fuction code ” **lib_path = "./ SC10CommandLib_win32.dll "**” to “ **lib_path = "./ SC10CommandLib_x32.dll "**” in **SC10_COMMAND_LIB.py** file.

3.1. SC10_COMMAND_LIB Namespace Reference

3.1.1. SC10 Class Reference

Public Member Functions

- def **__init__** (self)
- def **open** (self, serialNo, nBaud, timeout)
- def **is_open** (self, serialNo)
- def **close** (self)
- def **set_baud_rate** (self, baud_rate)
- def **set_mode** (self, mode)
- def **toggle_enable** (self)
- def **set_open_time** (self, time)
- def **set_close_time** (self, time)
- def **set_trigger_mode** (self, mode)
- def **set_external_trigger_mode** (self, mode)
- def **set_repeat_count** (self, repeat_count)
- def **get_baud_rate** (self, baud_rate)

- def **get_mode** (self, mode)
- def **get_enable_state** (self, enable_state)
- def **get_open_time** (self, time)
- def **get_close_time** (self, time)
- def **get_trigger_mode** (self, mode)
- def **get_external_trigger_mode** (self, mode)
- def **get_repeat_count** (self, repeat_count)
- def **get_closed_state** (self, state)
- def **get_interlock_tripped** (self, tripped_state)
- def **get_id** (self, id)
- def **save_settings** (self)
- def **store_configuration** (self)
- def **load_configuration** (self)

Static Public Member Functions

- def **list_devices** ()
- def **load_library** (path)

Data Fields

- **hdl**

Static Public Attributes

- None **sc10Lib** = None
- bool **isLoad** = False

3.1.2. Function Documentation

def close (self)

```
 Close opened SC10 device
Returns:
    0: Success; negative number: failed.
```

def get_baud_rate (self, baud_rate)

```
  Get baud rate
Args:
    baud_rate: 0: 9.6k, 1:115k
Returns:
    0: Success; negative number: failed.
```

def get_close_time (self, time)

```
  Get close duration
Args:
    time: the shutter's close time in ms
Returns:
    0: Success; negative number: failed.
```

def get_closed_state (self, state)

```
  Get closed state
Args:
    state: 1: shutter is closed, 0: shutter is open
Returns:
    0: Success; negative number: failed.
```

def get_enable_state (self, enable_state)

```
  Get State
Args:
    enable_state: 0: the shutter is disabled, 1: enabled
Returns:
    0: Success; negative number: failed.
```

def get_external_trigger_mode (self, mode)

```
  Get Ex trigger mode
Args:
    mode: external trigger mode
Returns:
    0: Success; negative number: failed.
```

def get_id (self, id)

```
  get the SC10 id
Args:
    id: output string (255)
Returns:
    0: Success; negative number: failed.
```

def get_interlock_tripped (self, tripped_state)

```
  Get interlock tripped
Args:
    tripped_state: 1: interlock is tripped, otherwise 0
Returns:
    0: Success; negative number: failed.
```

def get_mode (self, mode)

```
  Get operating mode
Args:
    mode: the mode value(1-5)
Returns:
    0: Success; negative number: failed.
```

def get_open_time (self, time)

```
  Get open duration
Args:
    time: the shutter's open time in ms
Returns:
    0: Success; negative number: failed.
```

def get_repeat_count (self, repeat_count)

```
  GReturn the repeat count
Args:
    repeat_count: repeat count, a value of 1-99
Returns:
    0: Success; negative number: failed.
```

def get_trigger_mode (self, mode)

```
  Get trigger mode
Args:
    mode: 0:internal trigger mode,1:external trigger mode
Returns:
    0: Success; negative number: failed.
```

def is_open (self, serialNo)

```
 Check opened status of SC10 device
Args:
    serialNo: serial number of SC10 device
Returns:
    0: SC10 device is not opened; 1: SC10 device is opened.
```

def list_devices () [static]

```
 List all connected mcm301 devices
Returns:
   The mcm301 device list, each deice item is serialNumber/COM
```

def load_configuration (self)

```
  Load configuration from EEPROM
Args:
Returns:
    0: Success; negative number: failed.
```

def load_library (path) [static]

def open (self, serialNo, nBaud, timeout)

```
 Open SC10 device
Args:
    serialNo: serial number of SC10 device
    nBaud: the bit per second of port
    timeout: set timeout value in (s)
Returns:
    non-negative number: hdl number returned Successful; negative number: failed.
```

def save_settings (self)

```
  Save current baud rate and output trigger mode
Args:
Returns:
    0: Success; negative number: failed.
```

def set_baud_rate (self, baud_rate)

```
 set SC10's serial baud rate
Args:
    baud_rate: SC10 baud rate, 0 for 9.6k and 1 for 115k
Returns:
    0: Success; negative number: failed.
```

def set_close_time (self, time)

```
 set close duration
```

```
Args:
    time: shutter's close time in ms
Returns:
    0: Success; negative number: failed.
```

def set_external_trigger_mode (self, mode)

```
 Set Ex Trigger mode
Args:
    mode: 0:set the output trigger to follow the shutter output when the SH05 is connected,
           1: force the trigger output to follow the controller output when an SH05 is equipped
Returns:
    0: Success; negative number: failed.
```

def set_mode (self, mode)

```
 Set operating mode
Args:
    mode: SC10 mode. 1-manual, 2-auto, 3-single, 4-repeat, 5-external gate
Returns:
    0: Success; negative number: failed.
```

def set_open_time (self, time)

```
 set open duration
Args:
    time: shutter's open time in ms
Returns:
    0: Success; negative number: failed.
```

def set_repeat_count (self, repeat_count)

```
 Set repeat count
Args:
    repeat_count: set the repeat count when in repeat mode, a value of 1-99
Returns:
    0: Success; negative number: failed.
```

def set_trigger_mode (self, mode)

```
 set the trigger mode
Args:
    mode: 0:internal trigger mode,1:external trigger mode
Returns:
    0: Success; negative number: failed.
```

def store_configuration (self)

```
  Store configuration, save current settings(ex. mode, open time, close time) into EEPROM
Args:
Returns:
    0: Success; negative number: failed.
```

def toggle_enable (self)

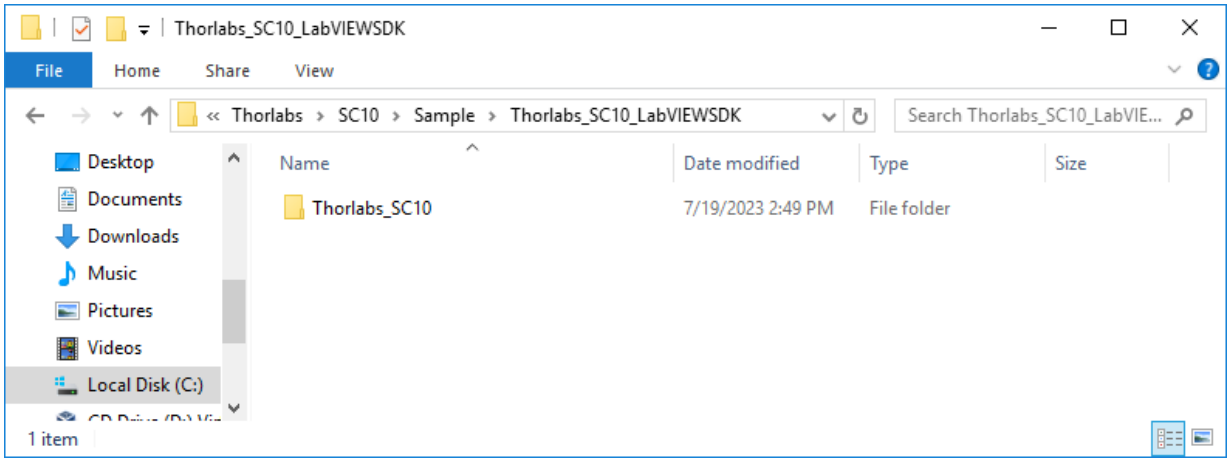
```
 Enable/Disable the shutter
Args:
Returns:
    0: Success; negative number: failed.
```

Chapter 4 LabVIEW Software Development Kit

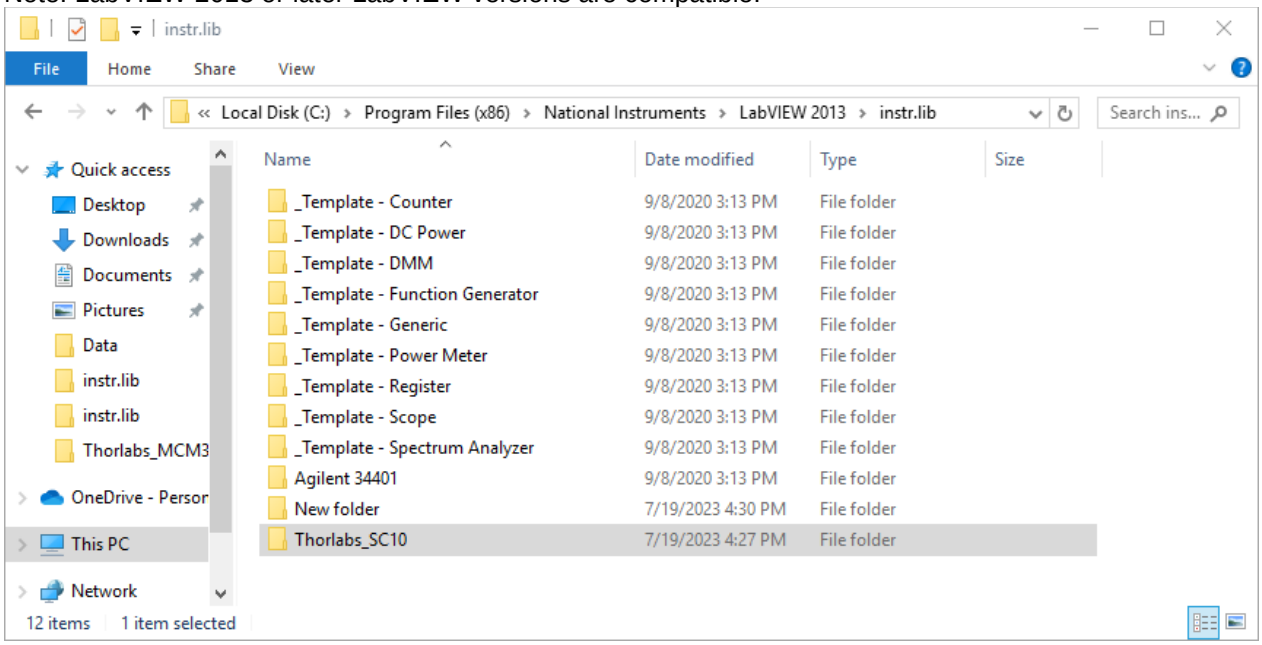
The user can start software development with LabVIEW 2013 or later versions based on LabVIEW instrument driver mechanism. The supported files are in **\Thorlabs_SC10_LabVIEWSDK** under the **Sample** directory.

How to install

Copy to instr.lib folder under LabVIEW installation folder.

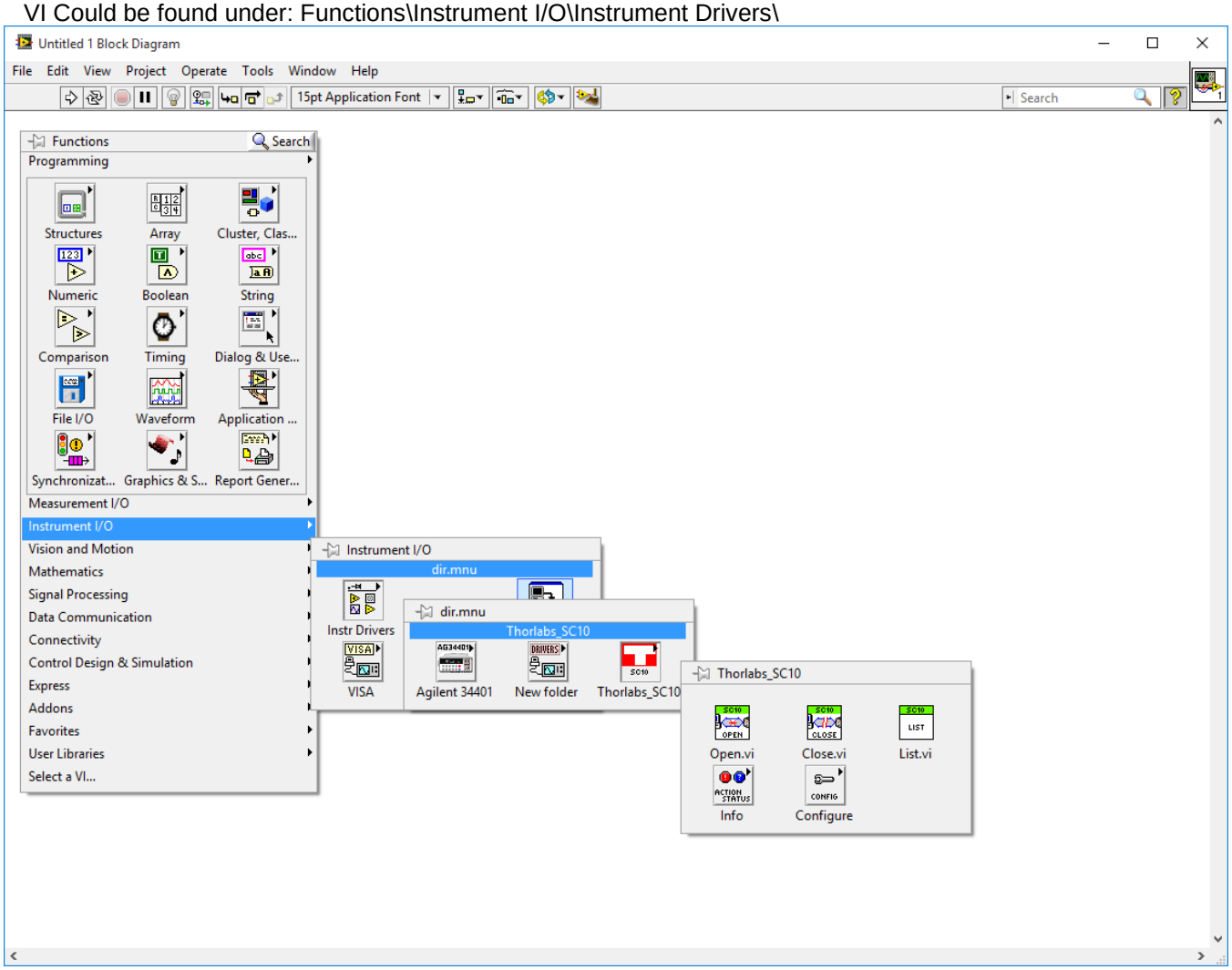


Destination folder: under %LabVIEW install path%\instr.lib
Typically, C:\Program Files (x86)\National Instruments\LabVIEW 2013\instr.lib
Note: LabVIEW 2013 or later LabVIEW versions are compatible.



How to find VI

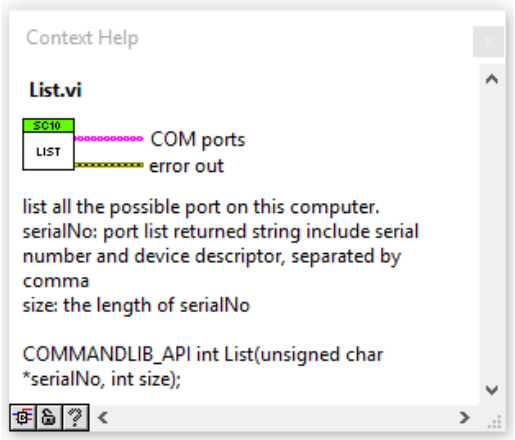
VI Could be found under: Functions\Instrument I/O\Instrument Drivers\



How to use

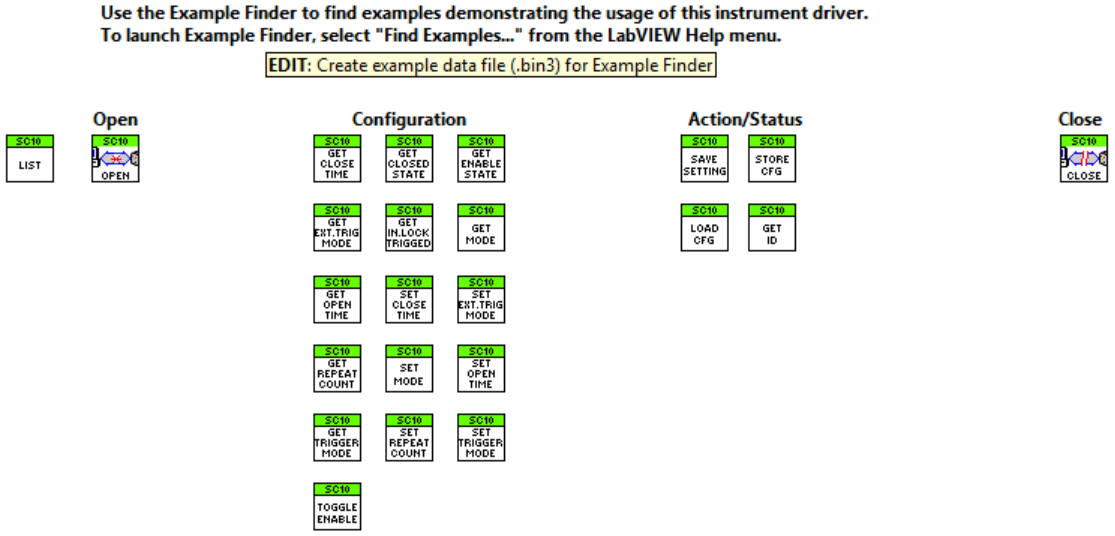
1. From VI

Note: Before you open the SDK LabVIEW project, make sure the device has been connected to the computer.



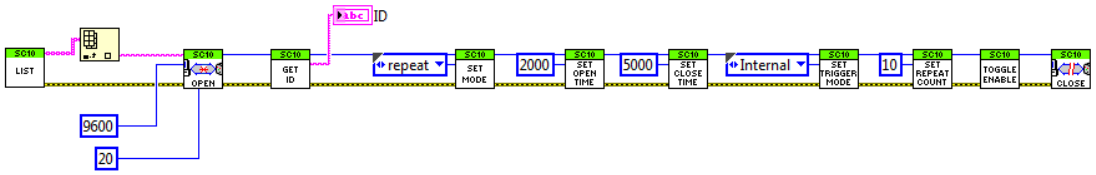
2. From VI tree

Some classic data flow in VI tree.



3. From example

Examples show the classic usage. Examples' path: instr.lib\Thorlabs_SC10\Example




UNLEASHED