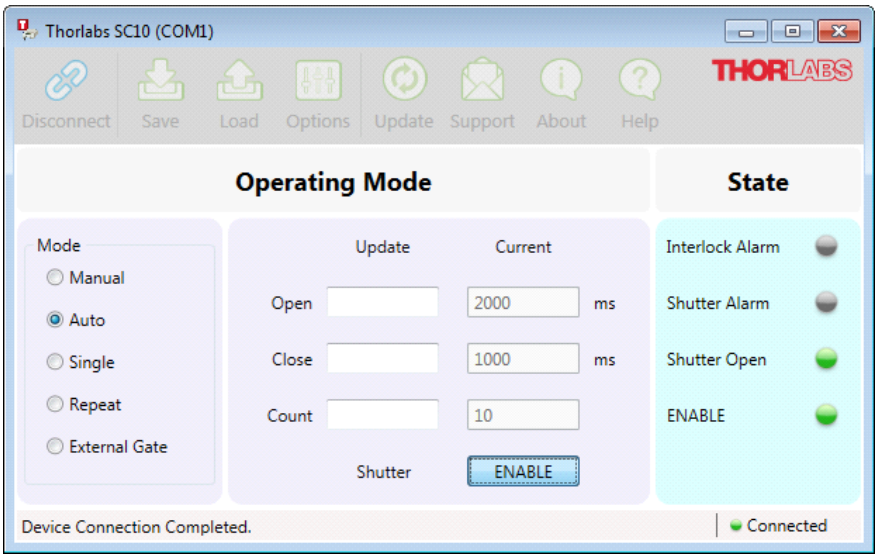
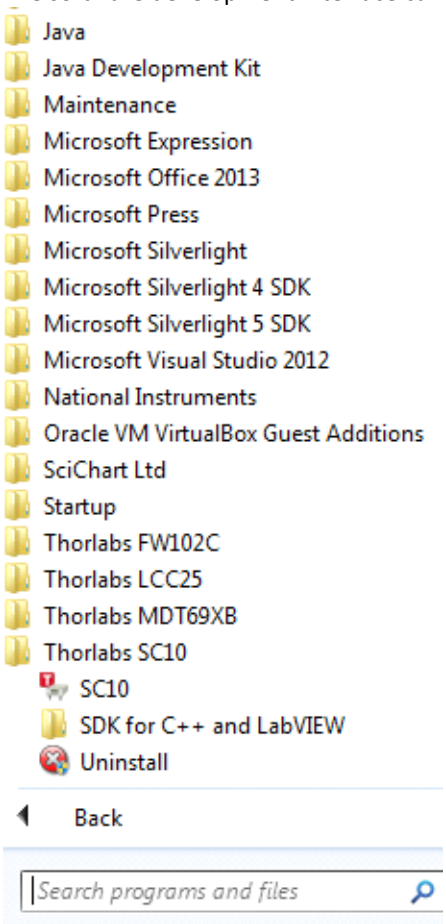
SC10

SDK Manual

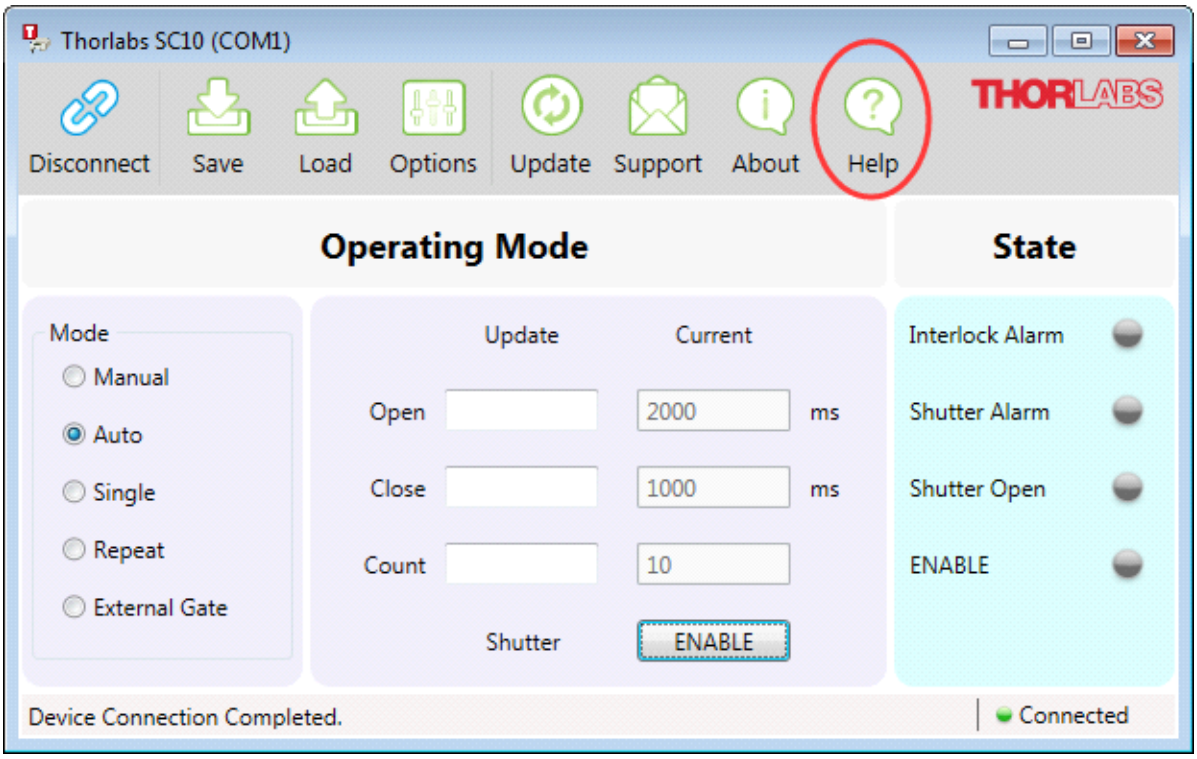


Software Development

User can start software development in C/C++ develop environment and LabVIEW.
The software development interface can be found in the start menu.



Or in the “Help” menu of the application.



- **Sample button on UI**

In this directory, you will find two zip files for software development, as shown below.

Name	Date modified	Type	Size
SC10 C++ SDK.zip	2017/8/1 17:13	WinRAR ZIP 压缩...	119 KB
SC10 LabVIEW Instrument Driver.zip	2017/8/1 17:13	WinRAR ZIP 压缩...	369 KB

Software Development (C/C++)

User can start software development with SC10CommandLib_win32.dll in C/C++ development environment which can be found in SC10 C++ SDK.zip. The corresponding header file is sc10_cmd_library.h.
Copy SC10CommandLib_win32.dll to your program folder, and make sure the library file and exe file are in the same folder.
You can download the user manual from our website to check the description of all the commands supported.

Commands and Queries

sc10_cmd_library.h File Reference

Functions

DllExport int **List** (char *serialNo)
DllExport int **GetPorts** (char *serialNo)
DllExport int **Open** (char *serialNo, int nBaud, int timeout)
DllExport int **IsOpen** (char *serialNo)
DllExport int **Close** (int hdl)
DllExport int **Read** (int hdl, char *b, int limit)
DllExport int **Write** (int hdl, char *b, int size)
DllExport int **Set** (int hdl, char *c, int size)
DllExport int **Get** (int hdl, char *c, char *d)
DllExport int **SetTimeout** (int hdl, int timeout)
DllExport int **SetBaudRate** (int hdl, int value)
DllExport int **SetMode** (int hdl, int value)
DllExport int **ToggleEnable** (int hdl)
DllExport int **SetOpenTime** (int hdl, int value)
DllExport int **SetCloseTime** (int hdl, int value)
DllExport int **SetTriggerMode** (int hdl, int value)
DllExport int **SetExternalTriggerMode** (int hdl, int value)
DllExport int **SetRepeatCount** (int hdl, int value)
DllExport int **GetBaudRate** (int hdl, int &value)
DllExport int **GetMode** (int hdl, int &value)
DllExport int **GetEnableState** (int hdl, int &value)
DllExport int **GetOpenTime** (int hdl, int &value)
DllExport int **GetCloseTime** (int hdl, int &value)
DllExport int **GetTriggerMode** (int hdl, int &value)
DllExport int **GetExternalTriggerMode** (int hdl, int &value)
DllExport int **GetRepeatCount** (int hdl, int &value)
DllExport int **GetClosedState** (int hdl, int &value)
DllExport int **GetInterlockTripped** (int hdl, int &value)
DllExport int **GetId** (int hdl, char *d)
DllExport int **GetCommands** (int hdl, char *d)
DllExport int **SaveSettings** (int hdl)
DllExport int **StoreConfiguration** (int hdl)
DllExport int **LoadConfiguration** (int hdl)

Function Documentation

DllExport int Close (int *hdl*)

close current opend port

Parameters:

hdl	handle of port.

Returns:

0: success; negtive number : failed.

DllExport 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;

DllExport 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;

DllExport 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;

DllExport 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;

DllExport 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;

DllExport 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;

DllExport 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;

DllExport 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;

DllExport 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;

DllExport 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.

DllExport 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;

DllExport 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;

DllExport 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.

DllExport 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.

DllExport 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.

DllExport 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.

DllExport 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.

DllExport 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;

DllExport 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;

DllExport 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;

DllExport 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;

DllExport 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;

DllExport 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;

DllExport 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;

DllExport 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;

DllExport 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.

DllExport 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:

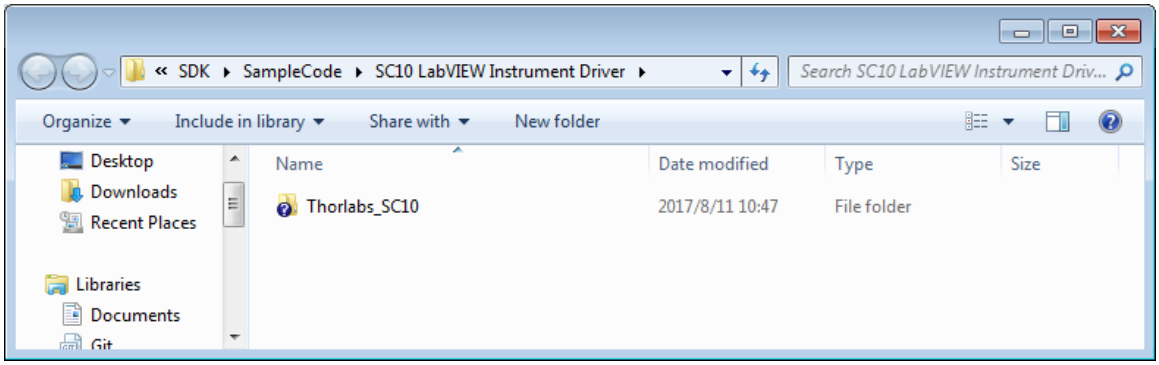
0: success;
0xEA: CMD_NOT_DEFINED;
0xEB: time out;

Software Development (LabVIEW)

User can start software development with LabVIEW 2011 (32bit/64bit) or later versions. The supported files are in \LabVIEW under the Sample directory.

- **How to install**

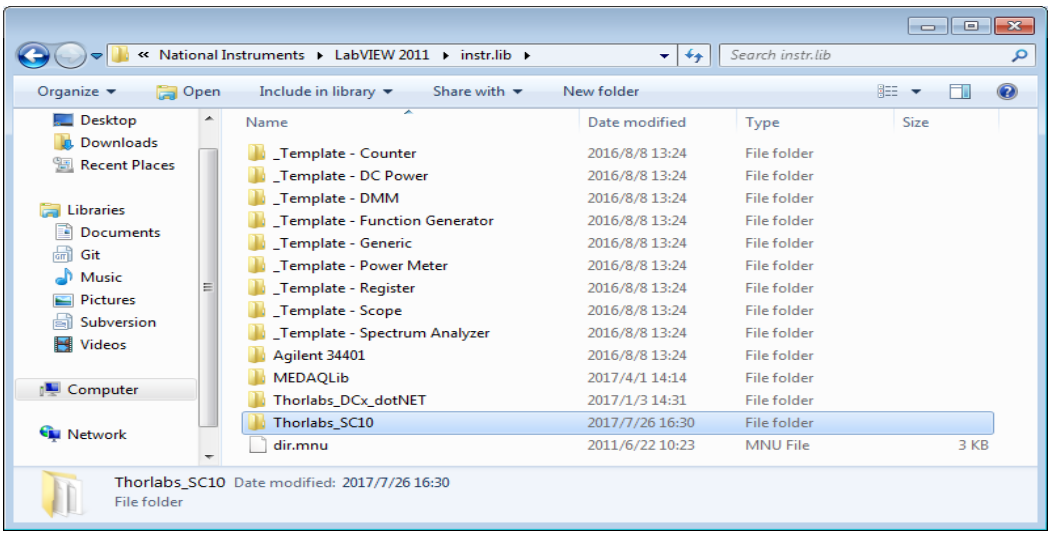
Unzip the zip file and copy the Thorlabs_SC10 and Thorlabs_uart folder to instr.lib folder under LabVIEW installation folder.
Thorlabs_SC10: device SDK in use based on Thorlabs_uart.



- ***Unzip folder***

- Destination folder: under %LabVIEW install path%\instr.lib

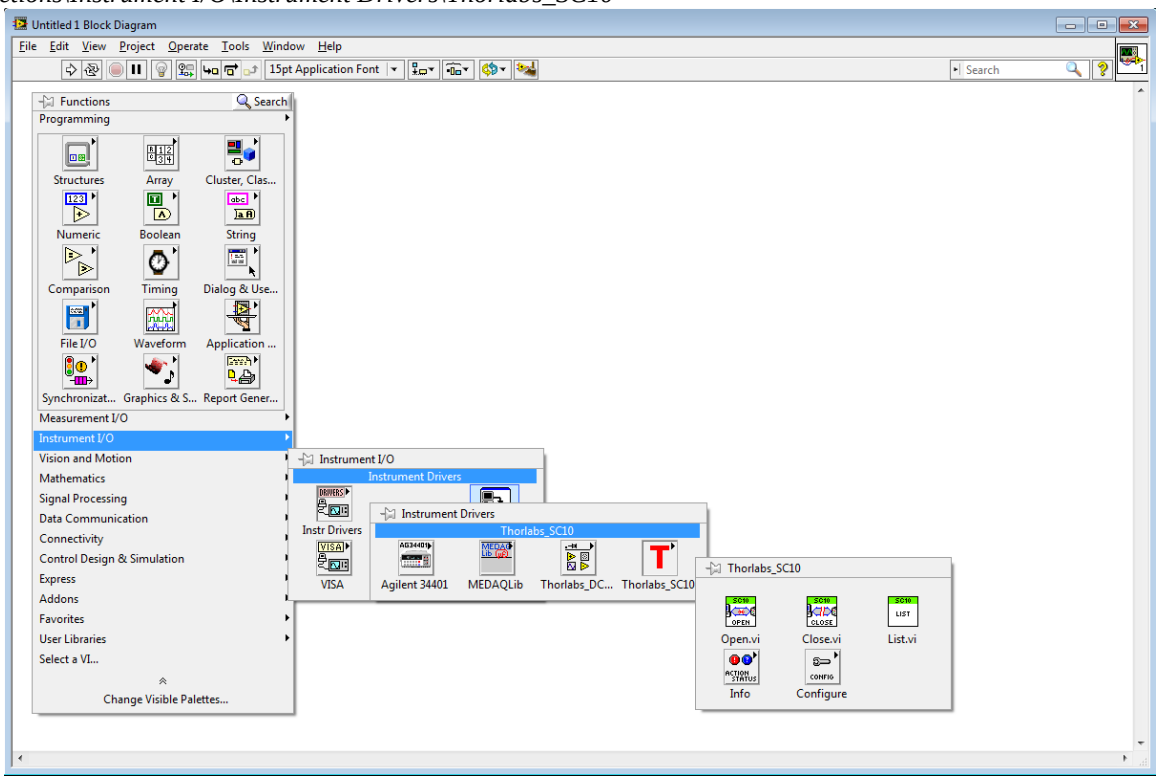
Typically, C:\Program file\ National instruments\LabVIEW 2011\instru.lib
and Later LabVIEW version are compatible.



- ***Destination folder***

- **How to find VI**

VI Could be found in right-click function palette under:
Functions\Instrument I/O\Instrument Drivers\Thorlabs_SC10



- **How to use**

Several examples can cover classic usage. Easy programming and detailed comment will help.
Typical examples path *C:\Program Files (x86)\National Instruments\LabVIEW 2011\instr.lib\Thorlabs SC10\Examples*