MAX32660 Secure Bootloader In-Application Programming with Python® User Guide

UG7233, Rev 0; 9/20

Abstract

This user guide details how update the end-user software application in the MAX32660 through the in-application programming, plus how to program the host code into the MAX32630FTHR board. Details on the MAX32660 secure bootloader can be found in the **MAX32660 Secure Bootloader User Guide**.

Table of Contents

List of Figures

List of Tables

Introduction

This application note provides the instructions to program example host code into the MAX32630FTHR development platform. The document also gives details related to hardware setup and application programming by using the MAX32660 bootloader and example host code.

Note that the screenshots may differ according to the software versions, but the steps will be same.

System Requirements

To compile and program the MAX32660 bootloader code into the MAX32660-EVSYS, the minimum requirements are as follows:

- Windows® PC
  - Windows 10, Windows 7
  - OpenSSL
  - Maxim Toolchain Software (more information, including download and installation instructions, is in this document).
- MAX32660-EVSYS and micro-USB cable
- MAX32630FTHR and micro-USB cable
- MAX32625PICO evaluation kit (EV kit) and micro-USB cable
- Test wires to connect the MAX32660-EVSYS and MAX32630FTHR

Maxim Toolchain Installation

To install the Maxim Toolchain to your PC, use the following steps:

1. Download the Arm® Cortex® Toolchain **here**.
2. After downloading is complete, double-click **ARMCortexToolchain.exe** and use the default settings and select **Next** until finished.
3. Select **Install the Driver/Run it Anyway** when Windows says that it does not recognize the driver.
4. In the folder *C:\Maxim*, double-click on **updates.bat**.

   If **updates.bat** fails, it may be necessary to open it in a text editor and call the commands manually.

Programming the MAX32630FTHR

To program example host code into the MAX32630FTHR, use the following steps:

1. Connect the grey 10-pin connector to the MAX32630FTHR and the MAX32625PICO board.
2. Connect the micro-USB cable to the MAX32625PICO and the PC.
3. Connect the micro-USB cable to the MAX32630FTHR and the PC.

Figure 1. The MAX32630FTHR and MAX32625PICO board connection.

4. Wait a few minutes for the Windows driver to install, then verify that it is installed correctly.
   a. In the Windows 10 search box, type **Control Panel** (or for Windows 7, click **Control Panel** on the right side of the **Start Menu**). Either click **Hardware and Sound**, then **Device Manager**, or type **Device Manager** in the search box in the upper right.
   b. If the drivers have correctly installed, you should see one port listed as **mbed Serial Port** for the MAX32625PICO. Note the COM port number for the USB serial device.

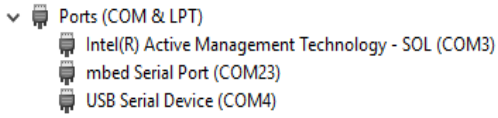


Figure 2. Serial port list.

   c. If you see the following, then you will need to install the correct Windows driver:

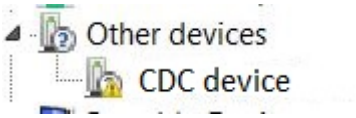


Figure 3. CDC device driver warning.

   d. Download the Arm Mbed® Windows serial port driver **here**.
   e. For Windows 10, run *mbedWinSerial_16466.exe* by double-clicking it.

f. For Windows 7:

i. Right-click on the *mbedWinSerial_16466.exe* file, and extract to a folder.

ii. Inside that folder, edit and add the following to the *mbedSerial_x64.inf* that the following italicized vid's and pid's are in the [_Devices] section.

[_Devices]
%S_DeviceDesc1%=Install,USB\VID_1F00&PID_2012&MI_01

%S_DeviceDesc1%=Install,USB\VID_1F00&PID_2012&MI_01

%S_DeviceDesc1%=Install,USB\VID_1F00&PID_2012&MI_01

%S_DeviceDesc1%=Install,USB\VID_1F00&PID_2012&REV0100

%S_DeviceDesc1%=Install,USB\VID_1F00&PID_2012

iii. Right-click on the **CDC device** warning, **Update Driver Software**, **Browse My Computer** for driver software, and enter the folder name from above. Wait at least 3 to 5 minutes for the driver to install.

iv. If there is still an issue, run the *mbed_xxx.exe* file.

5. In the **MinGW** window, navigate to the correct folder containing sample host binary image *max32630fthr-host-vx.x.bin*

6. In the **MinGW** window, type in the following:

openocd -s $MAXIM_PATH/share/openocd/scripts -f interface/cmsis-dap.cfg -f target/max3263x.cfg -c "program max32630fthr-host-vx.x.bin verify reset exit"

Alternatively, the .bin file may be dragged and dropped into the correct DAPLINK drive.

```
$ openocd -s $MAXIM_PATH/share/openocd/scripts -f interface/cmsis-dap.cfg -f target/max3263x.cfg -c "program max32630fthr-host-v2.3.bin verify reset;exit"
Open On-Chip Debugger 0.10.0+dev-snapshot (2018-12-21-12:38)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.org/doc/doxygen/bugs.html
none separate
Info : auto-selecting first available session transport "swd". To override use 'transport select <transport>'.
adapter speed: 2000 kHz
Info : CMSIS-DAP: SWD  Supported
Info : CMSIS-DAP: FW Version = 1.0
Info : CMSIS-DAP: Interface Initialised (SWD)
Info : SWCLK/TCK = 0 SWDIO/TMS = 1 TDI = 0 TDO = 0 nTRST = 0 nRESET = 0
Info : CMSIS-DAP: Interface ready
Info : clock speed 2000 kHz
Info : SWD DPIDR 0x2ba01477
Info : max32xxx.cpu: hardware has 6 breakpoints, 4 watchpoints
Info : Listening on port 3333 for gdb connections
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000b638 msp: 0x20080000
sp (/32): 0x0000B639
pc (/32): 0x0000B639
** Programming Started **
auto erase enabled
wrote 163840 bytes from file max32630fthr-host-v2.3.bin in 6.777226s (23.608 KiB/s)
** Programming Finished **
** Verify Started **
verified 162860 bytes in 0.568496s (279.761 KiB/s)
** Verified OK **
** Resetting Target **
Warn : Only resetting the Cortex-M core, use a reset-init event handler to reset any peripherals or configure hardware srst support.
```

Figure 4. Programming the MAX32630FTHR host.

7. Press the reset button on the MAX32630FTHR, as shown in **Figure 5**.

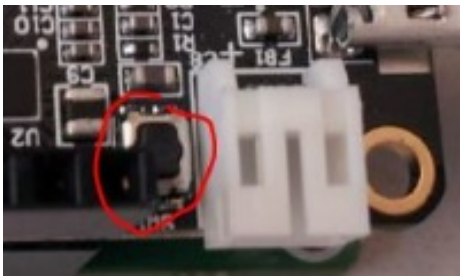


Figure 5. MAX32630FTHR host reset button.

8. Verify that the LED on the MAX32630FTHR is blinking, as shown in **Figure 6**.

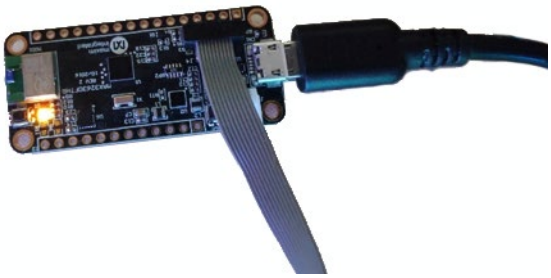


Figure 6. MAX32630FTHR host blinking LED.

Hardware Setup

Connect the MAX32630FTHR and MAX32660 with test wires according to **Table 1.** Pin diagrams for the MAX32630FTHR and MAX32660-EVSYS are given in **Figure 7** and **Figure 8**, respectively.

Table 1. Pin Connection between the MAX32630FTHR and MAX32660-EVSYS

PIN FUNCTION	MAX32660-EVSYS	MAX32630FTHR
EBL GPIO	P0.1	P5.4
I2C0_SCL	P0.2	P3.5 + 4.7K pullup
I2C0_SDA	P0.3	P3.4 + 4.7K pullup
RST	RSTN	P5.6

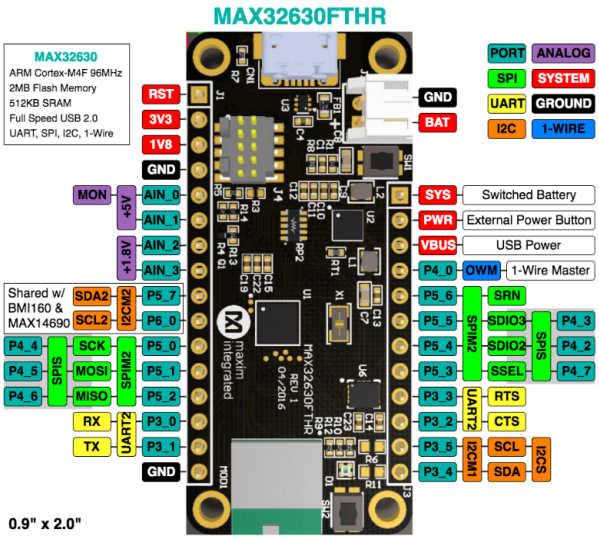


Figure 7. MAX32630FTHR pin diagram.

3rd	2nd	1st	NAME		NAME	1st	2nd	3rd
	CTS	SCK	P0_12		P0_10	MISO	TX	
	RTS	SSEL	P0_13		P0_11	MOSI	RX	
	TX	MISO	P0_4		P0_5	MOSI	RX	
TMR	SSEL	SDA	P0_3		P0_6	SCK	CTS	TX
32KCAL	SCK	SCL	P0_2		P0_7	SSEL	RTS	RX
RX	MOSI	SWCLK	P0_1		P0_8	SCL	SWDIO	
TX	MISO	SWDIO	P0_0		P0_9	SDA	SWCLK	
			RSTN		VCORE			
			GND		HDR_VIO			

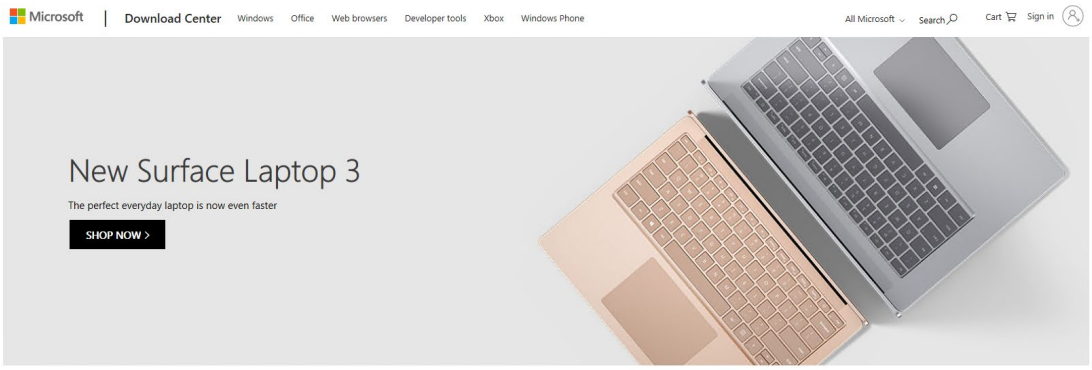
Figure 8. MAX32660-EVSYS pin diagram.

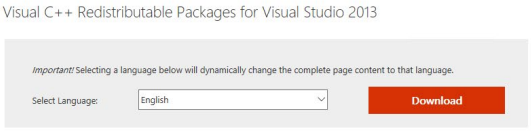
In-Application Programming

Installing Microsoft Visual C++ Runtime

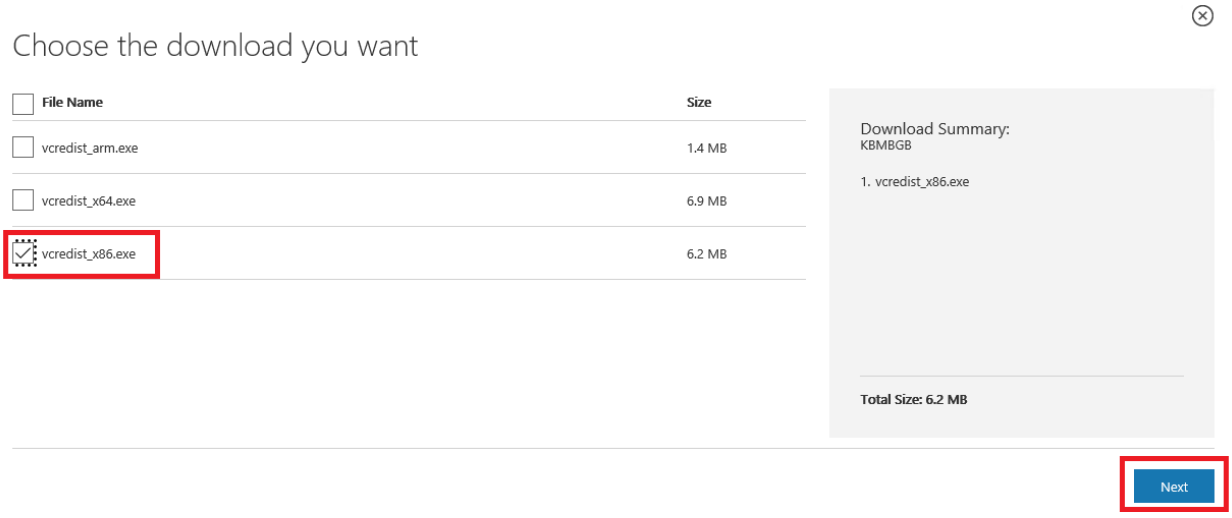
To download and install Microsoft Visual C++ Runtime, use the following steps:

1. Visit the download link **here** and click Download button.



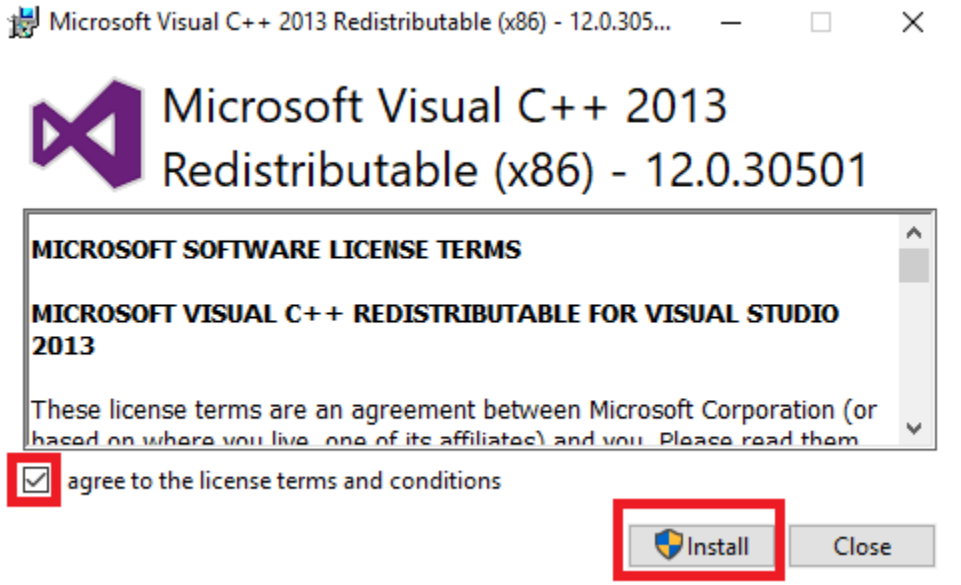


2. Select vcredist_x86.exe and continue by selecting **Next**.

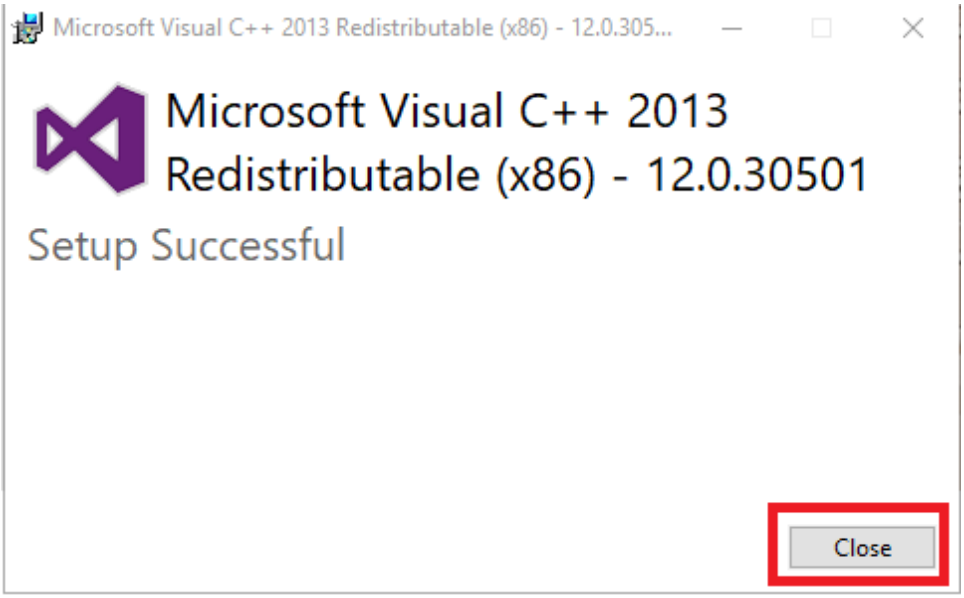


3. Download and run the setup file.

4. Read and click if you agree to the terms. Then select **Install**.



5. Close the installation application.



Installing OpenSSL Library

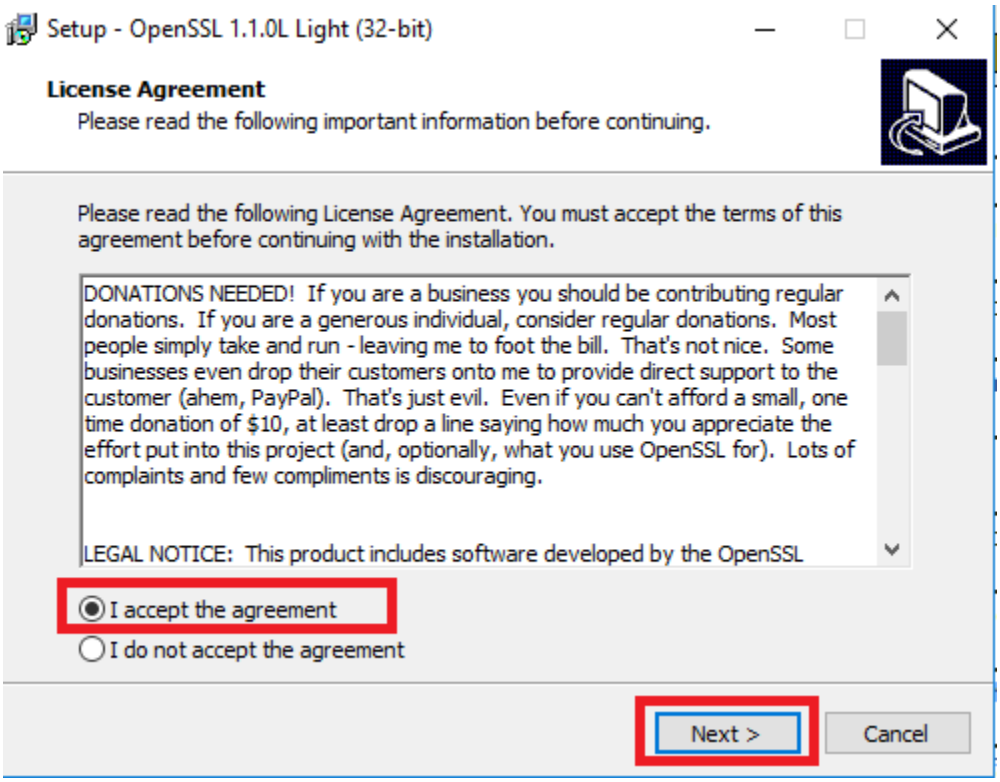
To download and install OpenSSL Library, use the following steps:

1. Visit OpenSSL downloads page **here**.
2. Select **Win32 OpenSSL v1.1.10L Light** package.

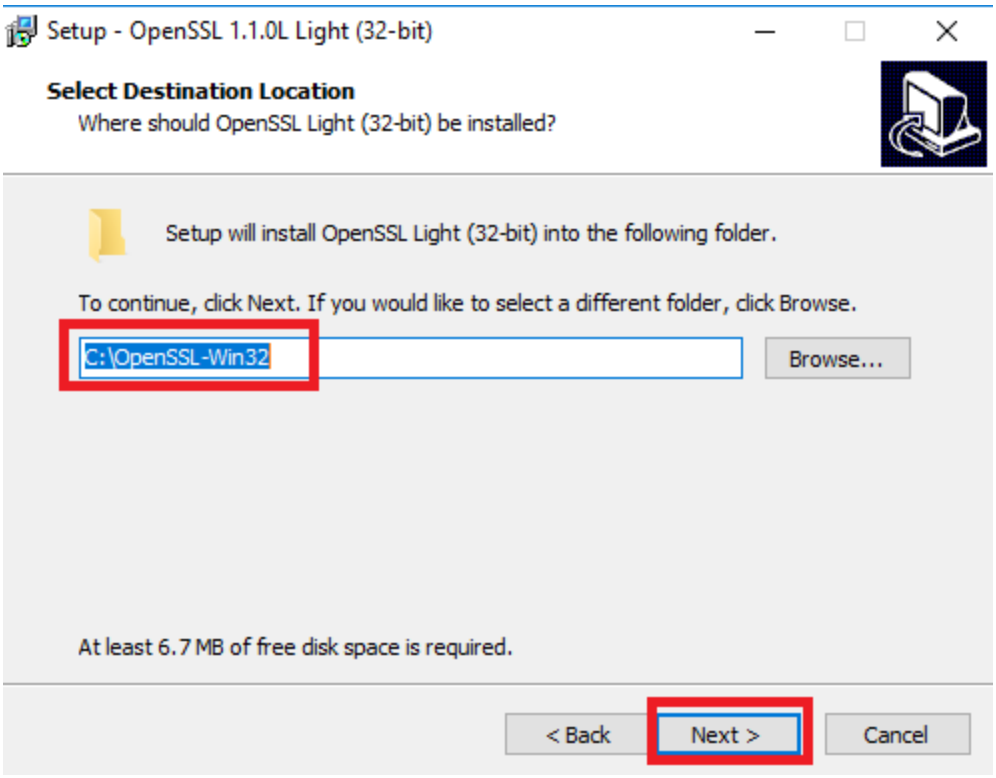
Download Win32/Win64 OpenSSL today using the links below!

File	Type	Description
Win64 OpenSSL v1.1.1f Light EXE | MSI (experimental)	3MB Installer	Installs the most commonly used build of OpenSSL and is subject t
Win64 OpenSSL v1.1.1f EXE | MSI (experimental)	63MB Installer	Installs Win64 OpenSSL v1.1.1f (subject to local and state laws. M
Win32 OpenSSL v1.1.1f Light EXE | MSI (experimental)	3MB Installer	Installs the most commonly used state laws. More information can
Win32 OpenSSL v1.1.1f EXE | MSI (experimental)	54MB Installer	Installs Win32 OpenSSL v1.1.1f (found in the legal agreement of th
Win64 OpenSSL v1.1.0L Light	3MB Installer	Installs the most commonly used build of OpenSSL and is subject t
Win64 OpenSSL v1.1.0L	37MB Installer	Installs Win64 OpenSSL v1.1.0L subject to local and state laws. M
Win32 OpenSSL v1.1.0L Light	3MB Installer	Installs the most commonly used state laws. More information can
Win32 OpenSSL v1.1.0L	30MB Installer	Installs Win32 OpenSSL v1.1.0L laws. More information can be fou
Win64 OpenSSL v1.0.2u Light	3MB Installer	Installs the most commonly used build of OpenSSL and is subject t
Win64 OpenSSL v1.0.2u	23MB Installer	Installs Win64 OpenSSL v1.0.2u subject to local and state laws. M
Win32 OpenSSL v1.0.2u Light	2MB Installer	Installs the most commonly used state laws. More information can
Win32 OpenSSL v1.0.2u	20MB Installer	Installs Win32 OpenSSL v1.0.2u laws. More information can be fou

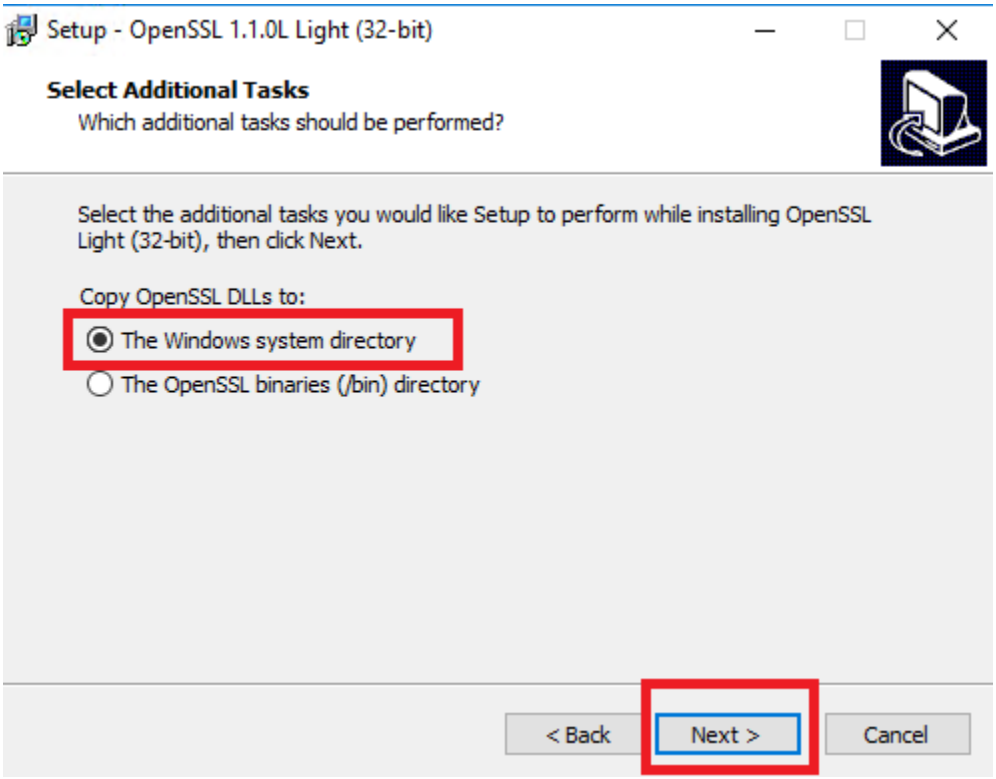
3. Download and run the installer.
4. Read and click if you agree to the terms. Then select **Next**.



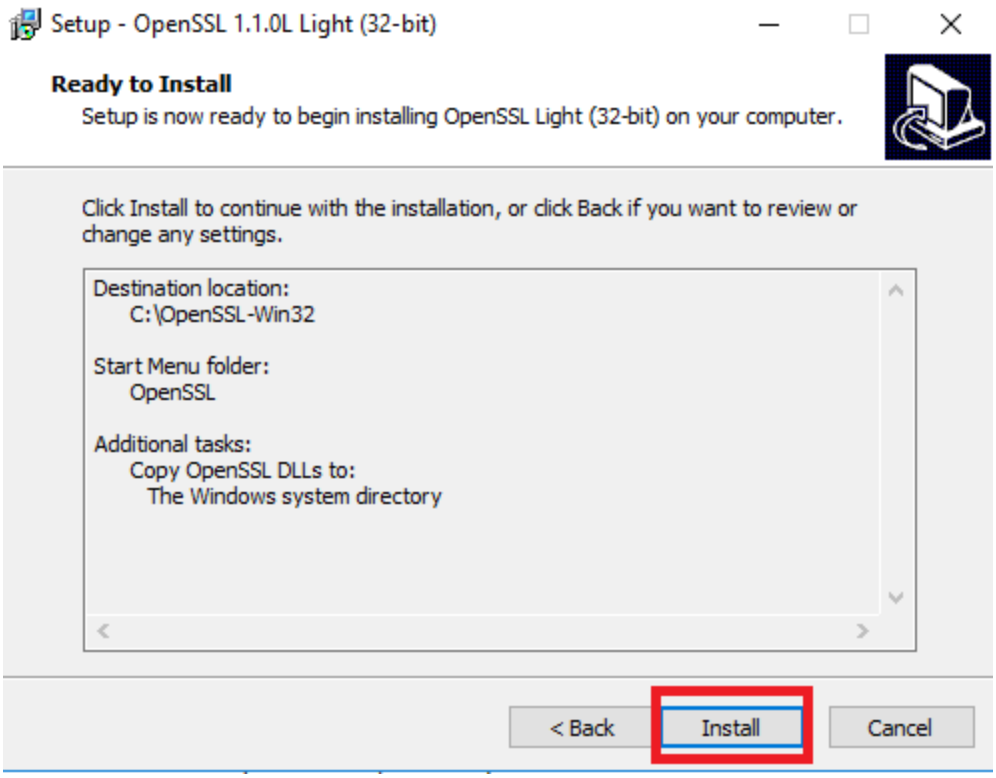
5. Leave the destination location default and click **Next**.



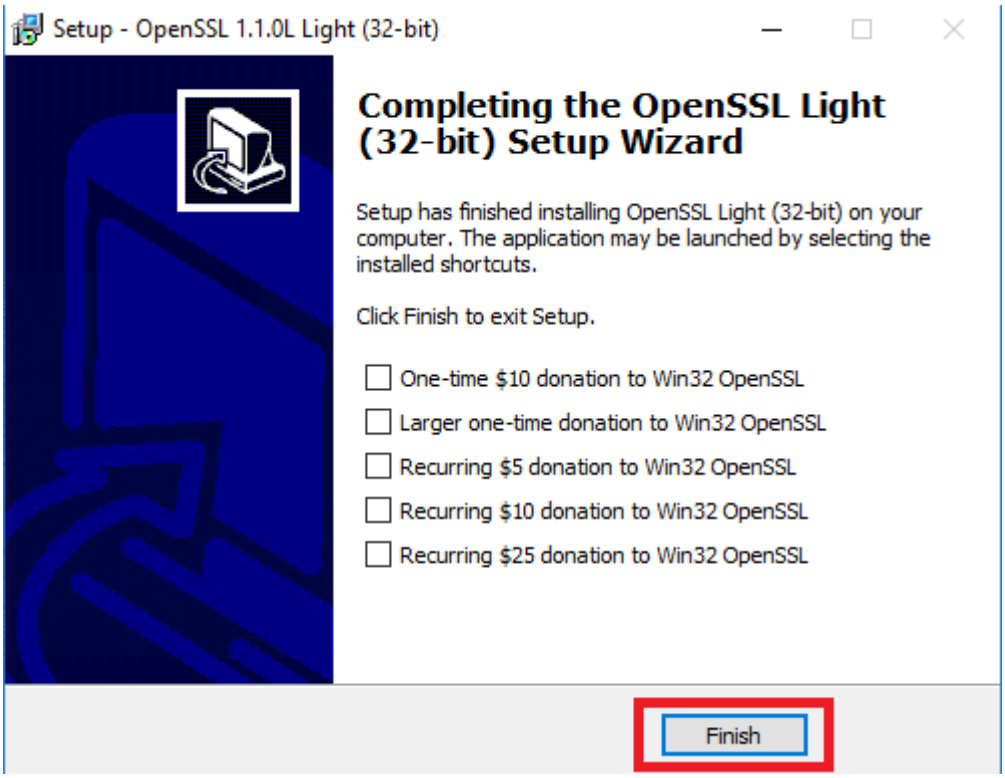
6. Leave "**The Windows system directory**" selected and click **Next**.



7. Click **Install**.



8. When the installation is completed, click **Finish**.



Installing Python

To download and install Python, use the following steps:

1. Download and install Python 2.7.13 **here**.
2. During installation, also install pip and add python to the path as seen in **Figure 9**.

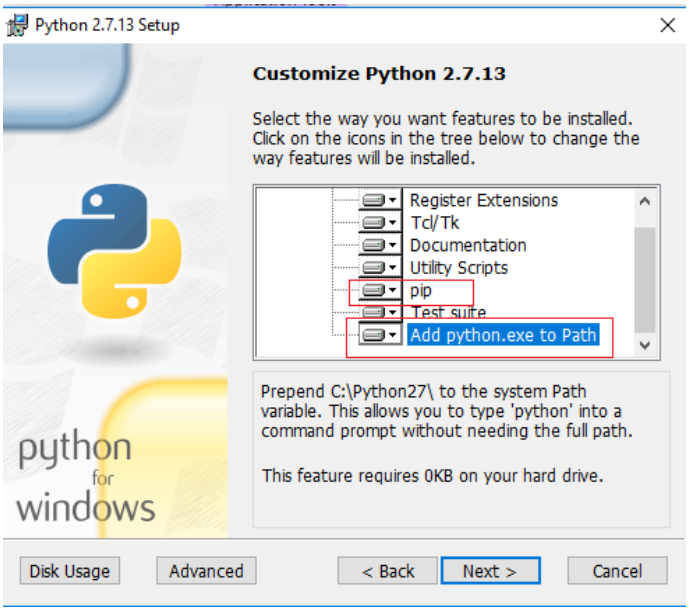


Figure 9. Python Installation

3. In the Windows search box, type **dos** and select the **Command Prompt**.

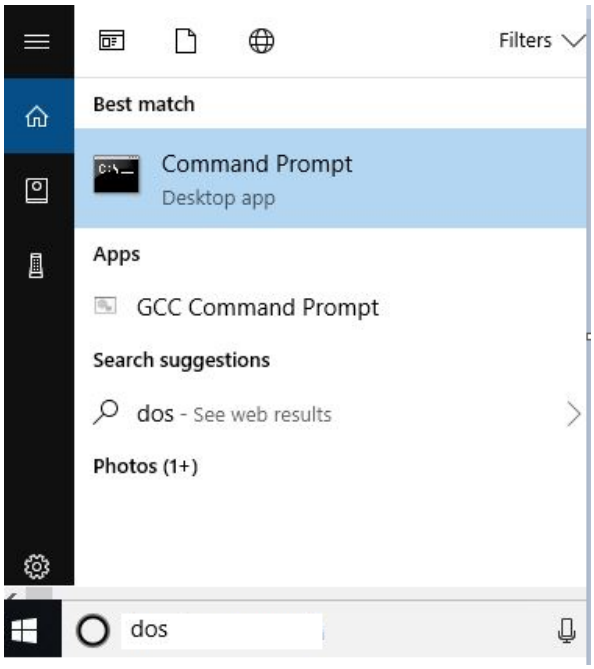


Figure 10. Navigating to the Command Prompt.

4. Add the Python components needed by typing the following at the DOS prompt:

pip install PySerial>=2.7

pip install colorama>=0.3.3

pip install enum34>=1.1.6

Compiling the Hello World Example with the Make Command

To compile Hello World with the make command, use the following steps:

1. Navigate to the correct directory consisting of the Hello World Example in the **MinGW** window with the following command:

cd "C:\gen_secure_bl_release_vx_x_x\examples\Hello_World"

2. Enter the following command in the **MinGW** window and wait several minutes for the command to complete:

make

3. After successful compiling, the Hello World binary image *max32660.bin* will be in the "*C:\gen_secure_bl_release_vx_x_x\examples\Hello_World\build"* directory.

Be sure that the correct linker file is used for generating the .bin file. A sample linker file, max32660.ld, can be found under the Hello_World example folder.

4. If you want to rebuild, then enter these commands, respectively:
make clean

make

Converting the .bin File to the .msbl File Format

The .msbl file is generated automatically by using a .msbl generator.

1. Navigate to the correct directory consisting of the msbl generator in the **MinGW** window with the following command:

cd "C:\gen_secure_bl_release_vx_x_x"

2. Enter the following command in the **MinGW** window window to convert the .bin application to a .msbl file:

msblGen.exe examples/Hello_World/build/max32660.bin MAX32660 8192 key.txt

3. Rename generated "*max32660.msbl" file as "Hello_World.msbl"*

In-Application Programming with Python

To flash the application to the MAX32660 by using the MAX32660 bootloader, use the following steps:

At the command prompt or in the PowerShell window, enter the following commands, replacing COMxx with the correct USB serial device COM port found in section Programming the MAX32630FTHR at step 4.b.

python ./download_fw_over_host.py -f "Hello_World.msbl" -p "COMxx" -d 2

```
C:\test\max32660_demo>python .\download_fw_over_host.py -f Hello_World.msbl -p COM7 -c i2c


MAXIM FIRMWARE DOWNLOADER 0.34


>>> Parameters <<<
Mass Flash:  False
Reset Target:  False
EBL mode:  0
Delay Factor:  1
Port:  COM7
MSBL/Binary input file:  Hello_World.msbl
Comm Interface:  i2c
COM7 is open...


Initializing bl downloader
Input file name: Hello_World.msbl
### Press double Ctrl + C to stop
msbl file name: Hello_World.msbl
magic: msbl  formatVersion: 0  target: MAX32660  enc_type:   numPages: 6  pageSize: 8192  crcSize: 4 size of header: 76
  resv0:  0
nonce : 00 00 00 00 00 00 00 00 00 00 00 00
auth : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
resv1 : 00 00 00
last_pos: 76
Total file size: 49328 CRC32: 0x2abd9fe4L
Reading msbl file succeed.

Bootloader communication interface as i2c
Command: set_cfg comm i2c

Set comm interface to i2c
In silent mode. ret: 0

Set timeout mode to enter bootloader
Command: set_cfg host ebl 0

Set ebl_mode to 0

Set delay factor in host
Set bl comm delay factor to 1

Downloading msbl file

Enable image on RAM:  False
CMD :image_on_ram 0

In image_on_ram Mode.
platform Bootloader_MAX32664
sensors
err 0

firmware_ver HSP2_3_2.3
hub_firm_ver 3.4.1

Get page size
Target page size: 8192

Get USN
USN = 0400134b0e801241ffffffacffffff78000000b000045d0

Set number of pages to download
Set page size(6) successfully.

Set IV
set_iv 00000000000000000000000

Set IV bytes succeed.

Set Auth
set_auth 00000000000000000000000000000000

Set Auth bytes succeed.

Erase App
Erasing App flash succeed.

Enter flashing mode
flash command succeed.
Flashing 1/6 page...[DONE]
Flashing 2/6 page...[DONE]
Flashing 3/6 page...[DONE]
Flashing 4/6 page...[DONE]
Flashing 5/6 page...[DONE]
Flashing 6/6 page...[DONE]
Flashing MSBL file succeed...

Jump to main application
Jumping to main application. ret: 0
SUCCEED...
Closing
```

Figure 11. Downloading firmware with the download_fw_over_host Python script.

5. The application has now been flashed to the chip. For advanced usage of flasher script, refer to the *Firmware_downloader_usage.txt* file.

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	9/20	Initial release	—