# MAX32663 Secure Bootloader In-Application Programming with Python® User Guide

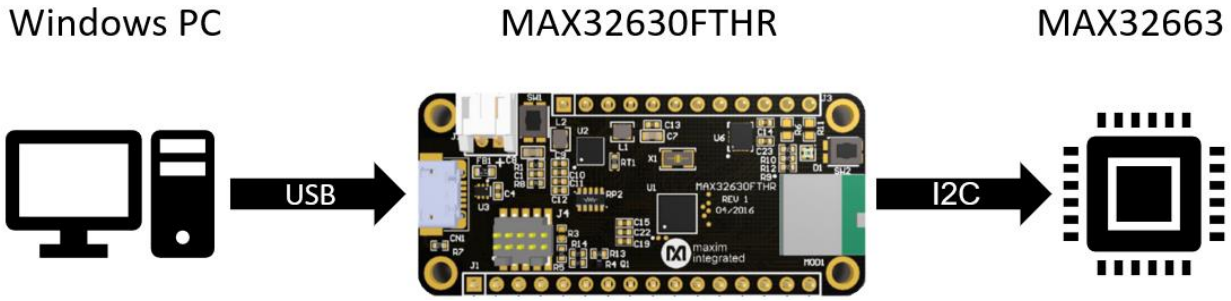

## Abstract

This guide describes how to securely update application firmware via the MAX32663 Secure Bootloader and a Python host. The protocol details for the MAX32663 Secure Bootloader can be found in the MAX32663 Secure Bootloader User's Guide.

## Table of Contents

## Introduction

The MAX32663 Secure Bootloader provides an I2C interface that facilitates the transfer of a keyed and encrypted firmware image from an I2C host to the internal flash. This document describes one method of using this interface to securely program a firmware image using the MAX32630FTHR and Python-based software running on a Windows PC Host.



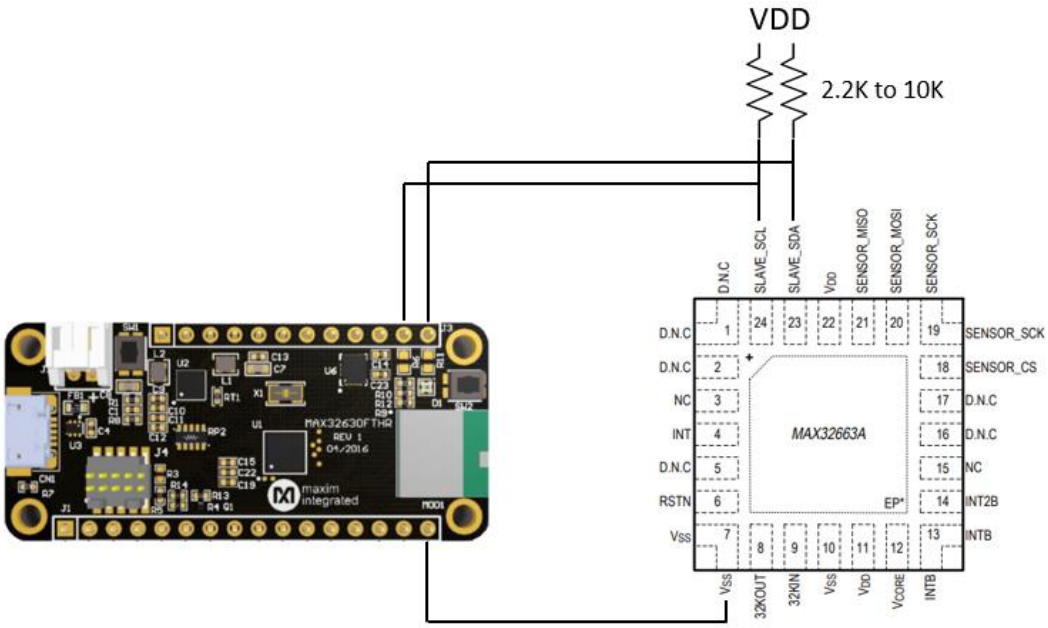

## MAX32630FTHR

The I2C bootloader interface on the MAX32663 can be mastered by any host I2C-capable micro and user firmware. The bootloader protocol is detailed in the MAX32663 Secure Bootloader User's Guide. However, for development and production, it is desirable that the preprogrammed MAX32630FTHR is included in the MAX30003WING2# evaluation kit.

The MAX32630FTHR contained in the MAX30003WING2# evaluation kit is used to program the MAX32663 on the kit's main PCB. The MAX32630FTHR can be removed from the main PCB and wired directly to the user's product. Note that the MAX32630FTHR is referenced to $V_{DD}$ = 3.3V. Targets referenced differently require voltage translation.

### MAX32630FTHR I2C Pin Connections

The MAX32630FTHR is connected to the target system as follows:




| MAX32630FTHR PIN | SIGNAL NAME | MAX32663 PIN |
|---|---|---|
| J3.11 | SCL | 24 |
| J3.12 | SDA | 23 |
| J2.16 | GND | 7 and 10 |

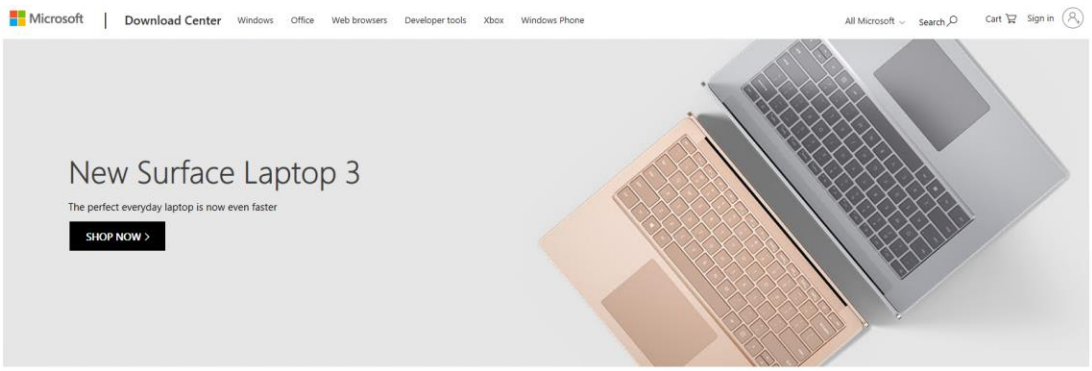
The target system must supply the power and not the MAX32630FTHR.

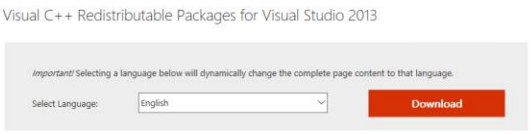
## Host Software

The host software uses Python. Several software prerequisites are required to run the Python script on Windows

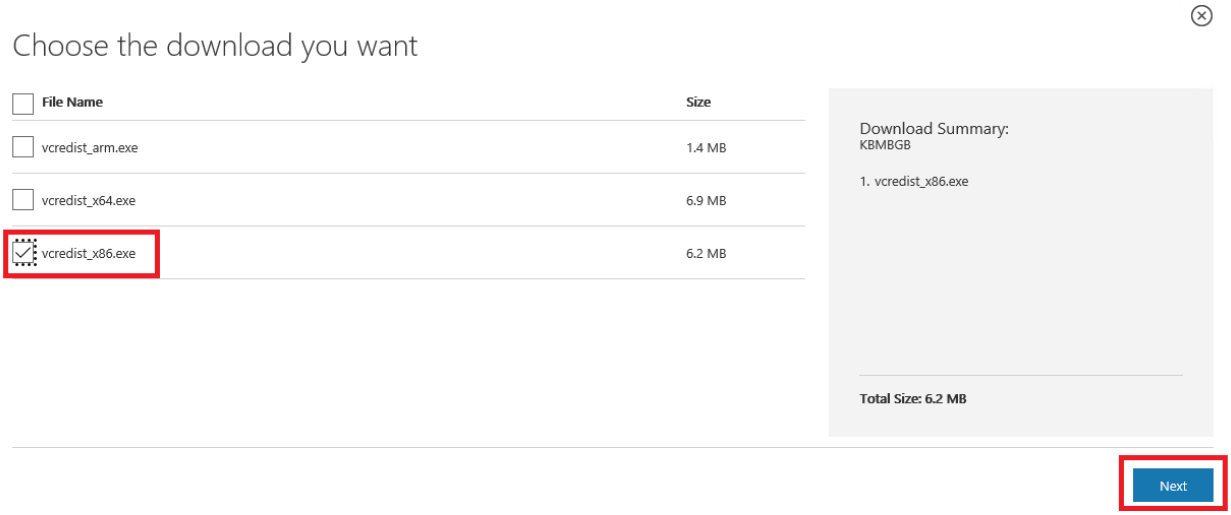
Use the following steps to download and install Microsoft Visual C++ Runtime:

1. Visit the download link **here.** Click **Download**.

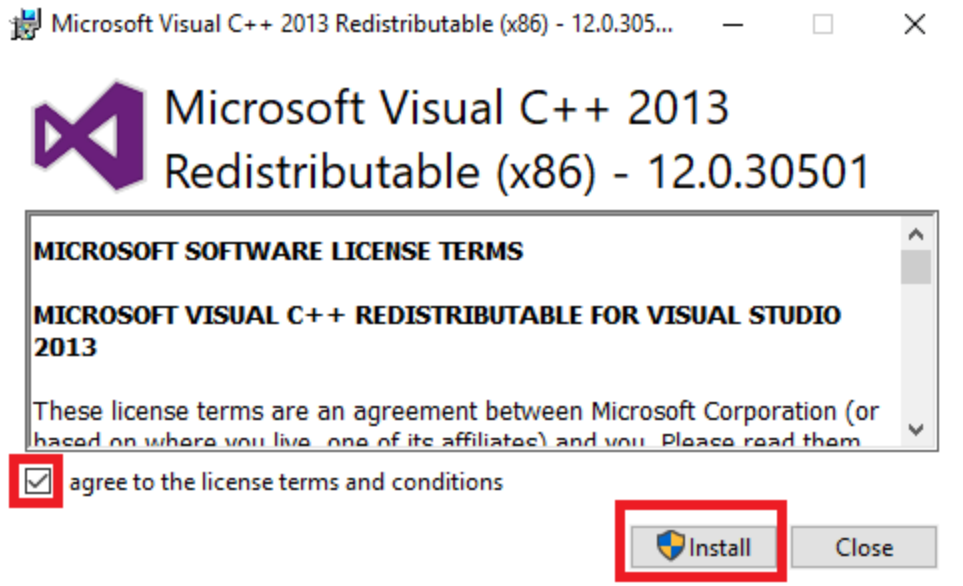


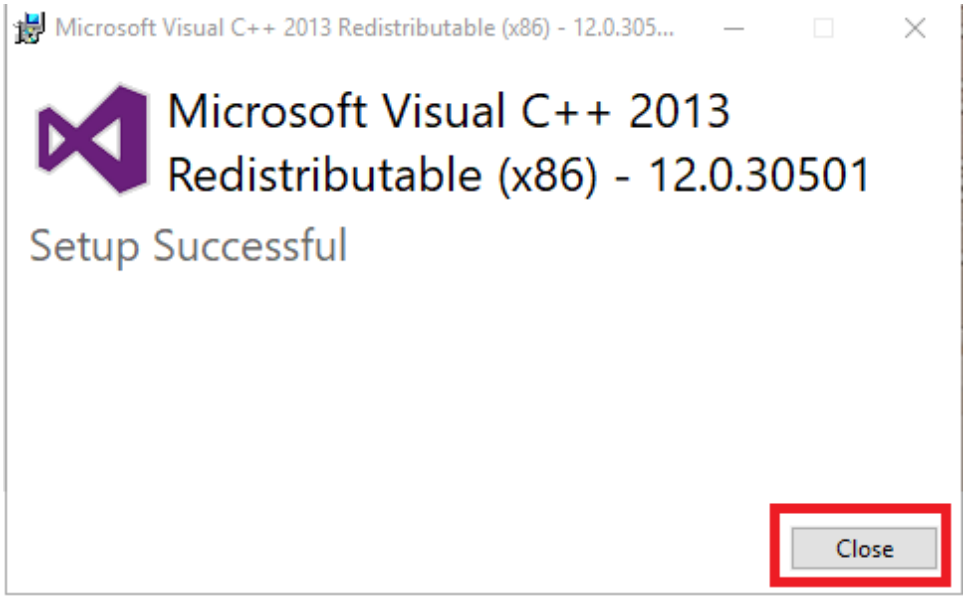

2. Select **vcredist_x86.exe.** Select **Next**.




3. Download and run the setup file.
4. Read and click to agree to the terms. Select **Install**.




5. Close the installation application.




### Installing the OpenSSL Library

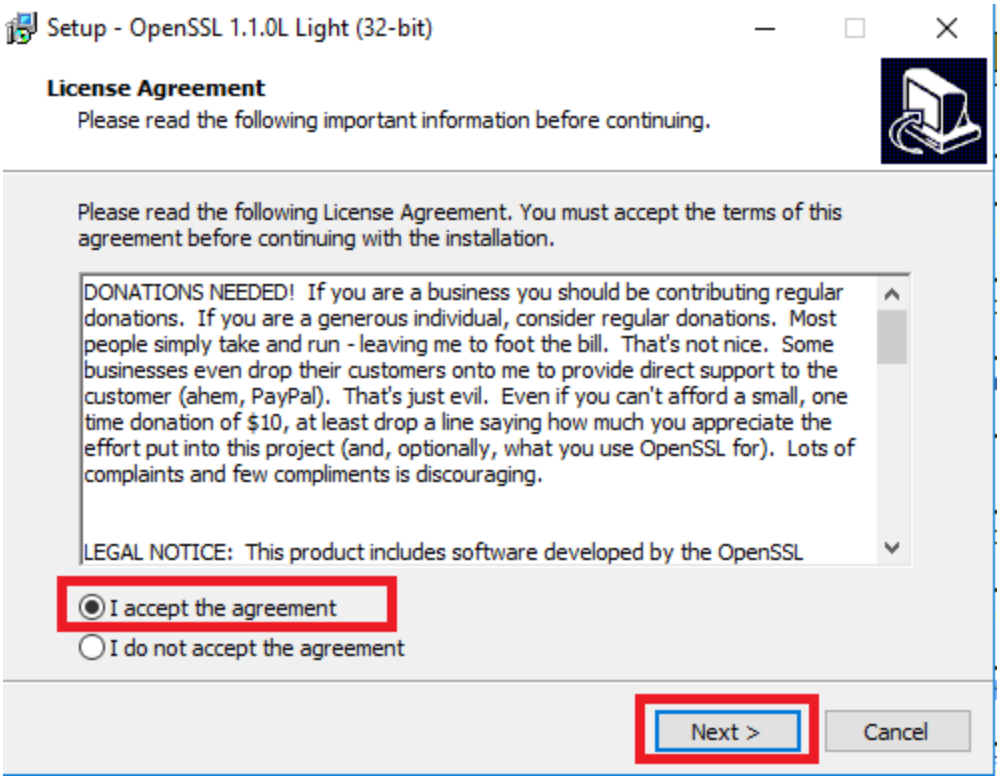
Use the following steps to download and install the OpenSSL Library:

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

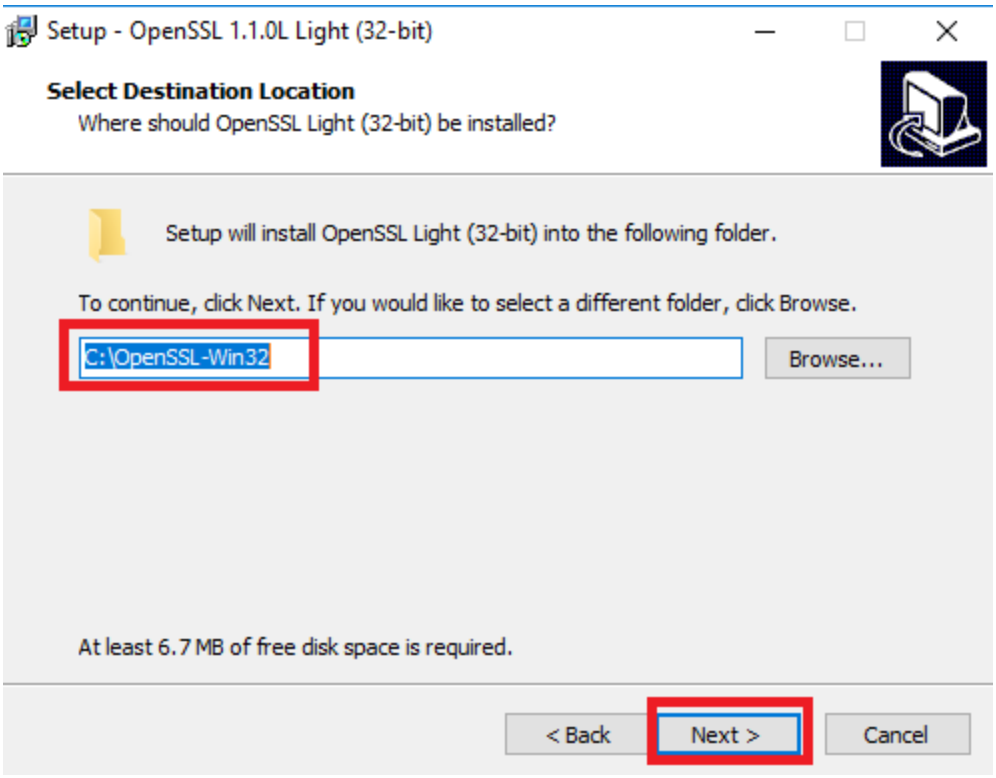
Download Win32/Win64 OpenSSL today using the links below!

| 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 |

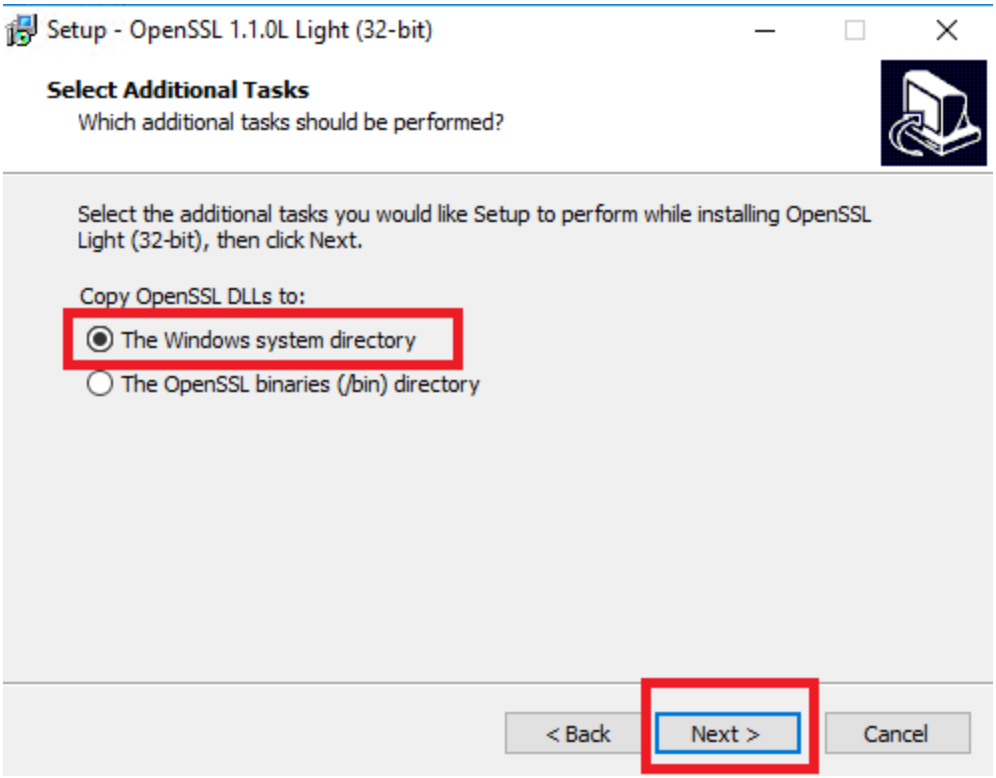
3. Download and run the installer.
4. Read and click to agree to the terms. Select **Next**.



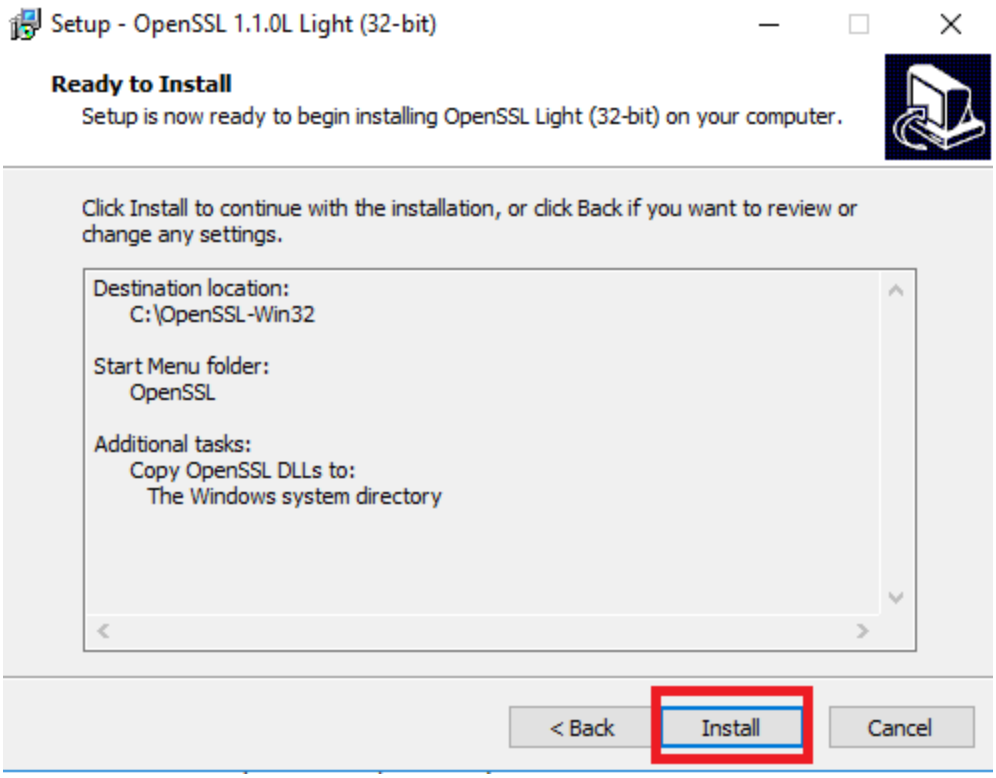

5. Leave the destination location default. Click **Next**.



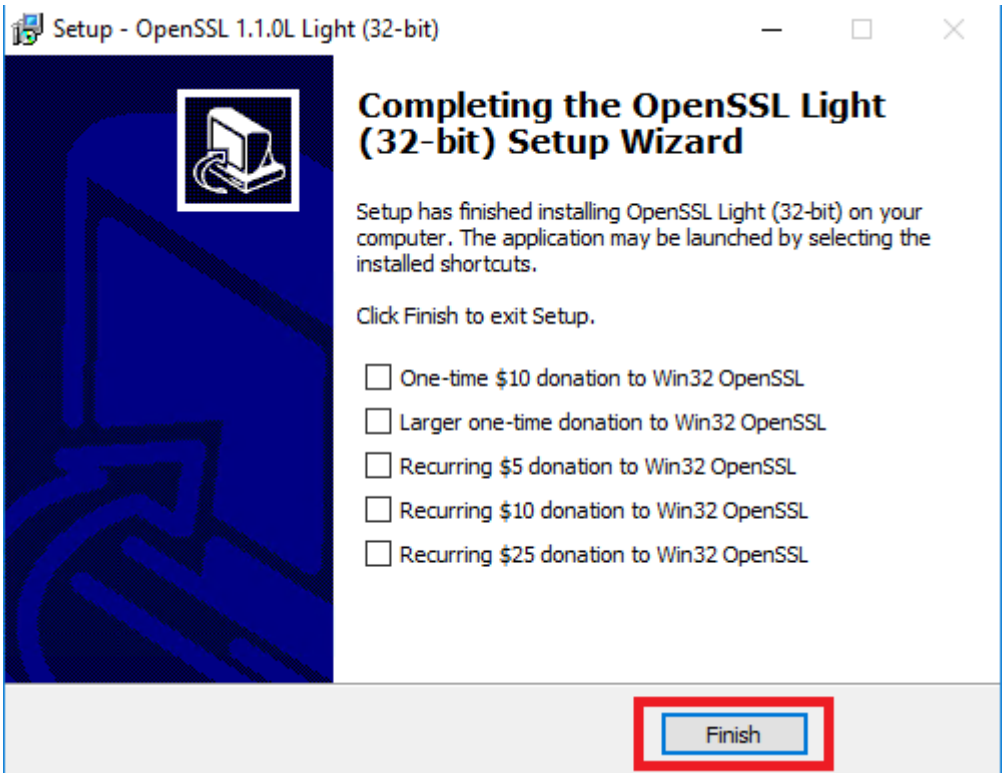

6. Leave **The Windows system directory**. Click **Next**.




7. Click **Install**.



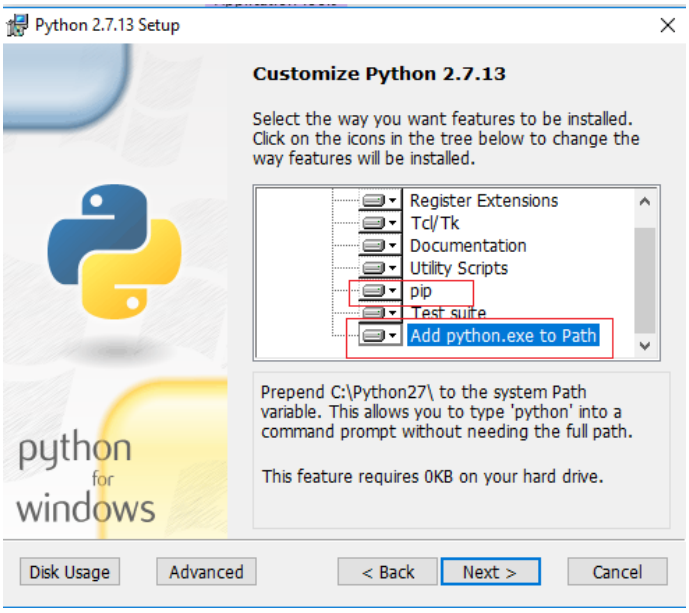

8. Click **Finish** when the installation is complete.



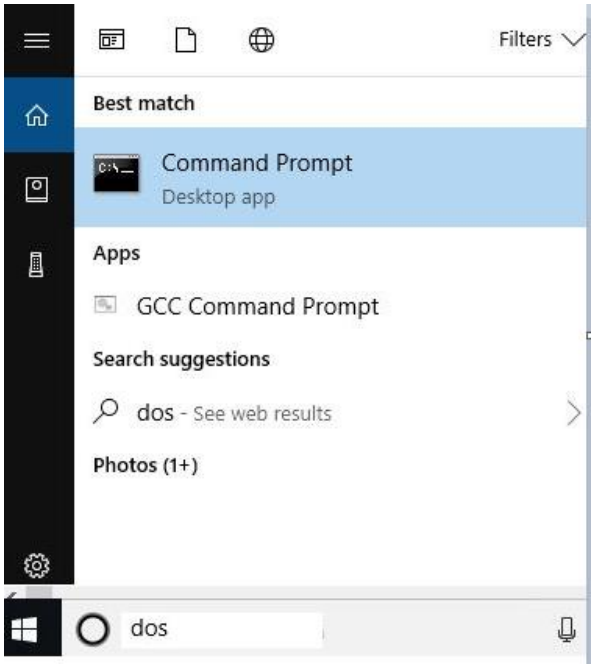
## Installing Python

Use the following steps to download and install Python:

1. Download and install Python 2.7.13 **here**.
2. Install **pip** and add **python** to the path during installation.




3. Type DOS in the Windows search box and select the Command Prompt.




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

a. pip install PySerial>=2.7

b. pip install colorama>=0.3.3

c. pip install enum34>=1.1.6

### In-Application Programming with Python

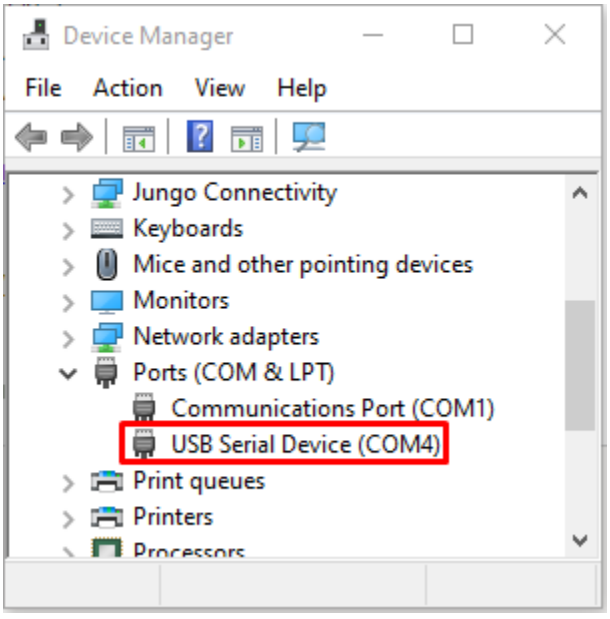
The bootloader requires a specific firmware image, which is provided and signed by the algorithm vendor. The signed firmware images have an **msbl** file extension. The msbl file is found along with other supporting software, including the download_fw_over_host.py python script, on the MAX32663 product page on the Design Resources tab.

Use the following steps to flash the application to the MAX32663:

Enter the following commands in the command prompt or in the PowerShell window, replacing COMxx with the COM port of the connected MAX32630FTHR. Replace FIRMWARE with the name of the encrypted firmware file provided by the algorithm vendor.

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

The COM port of the MAX32630FTHR is found by examining the **Ports (COM & LPT)** while hot plugging the MAX32630FTHR's USB connection to the host Windows PC. For example, the **USB Serial Device (COM4)** in the following image disappears when the MAX32630FTHR is disconnected from the PC and reappears when it is reconnected. The COM port number, four in this case, is likely to be different on each machine.



```
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 00000000000000000000

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
```

## Trademarks List

Microsoft is a registered trademark and registered service mark of Microsoft Corporation.

Windows is a registered trademark and registered service mark of Microsoft Corporation.

## Revision History

| REVISION NUMBER | REVISION DATE | DESCRIPTION | PAGES CHANGED |
|---|---|---|---|
| **0** | 10/20 | Initial release | — |
| | | | |
| | | | |
| | | | |
| | | | |