

# How to compile mbed platform

mbed CLI Compiler

# mbed CLI prerequisite

## Step 1 - Install Python, Git, Mercurial and GCC

### I. Install python

- 1) Go to download page for python (<https://www.python.org/downloads/>)
- 2) During installation, Add Python to system path
- 3) Wait for Python to install

### II. Install Git

- 1) Go to download page for git(<https://git-scm.com/download/win>)
- 2) Wait for Git to install

# mbed CLI prerequisite

## III. Install Mercurial

- 1) Go to download page for git(<https://www.mercurial-scm.org/downloads/>)
- 2) Wait for Git to install after click 'Add the installation path to the search path' during installation

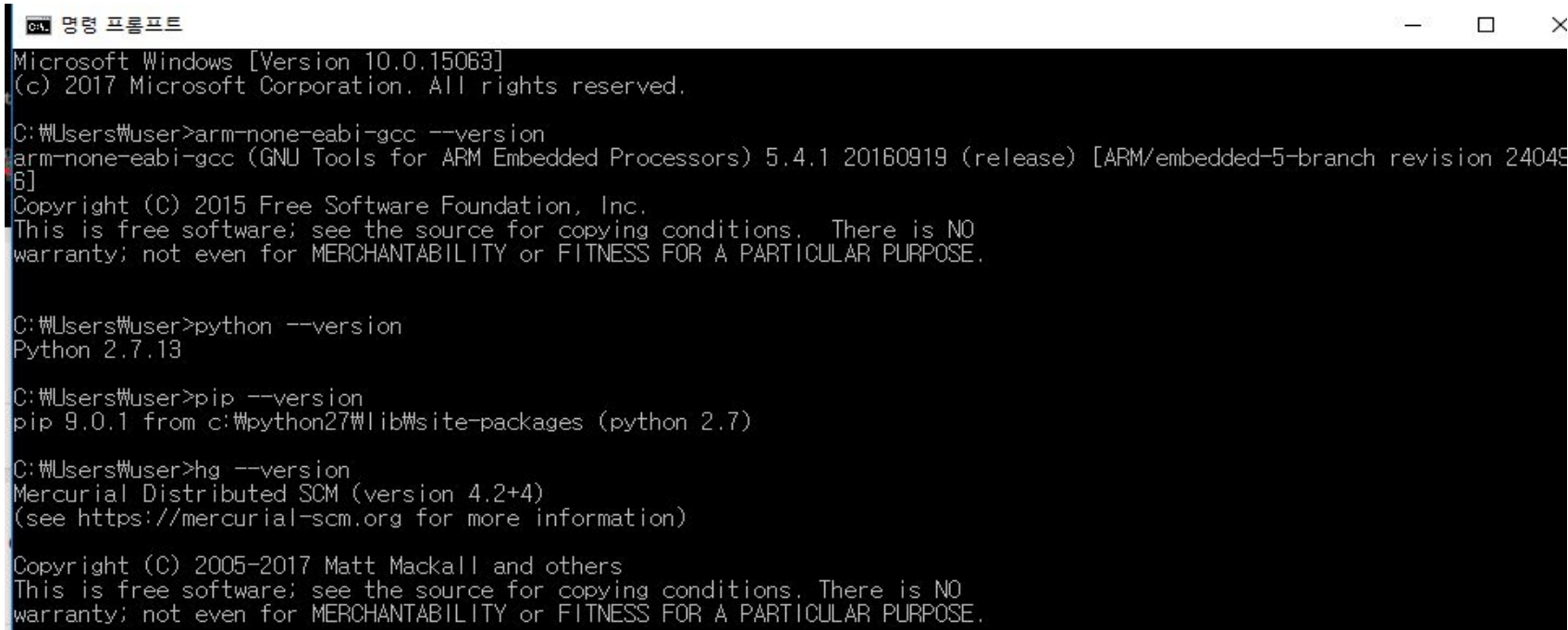
## IV. Install GCC

- 1) Go to download page(<https://launchpad.net/gcc-arm-embedded>) (GNU ARM Embedded Toolchain)
- 2) Wait for GCC to install after adding path to the environment, i.e., click 'Launch gccvar.bat' during installation

<http://d2.naver.com/helloworld/1011> (Comparison between Git and Mercurial)

# mbed CLI prerequisite

## Step 2 Verify installation



```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\User>arm-none-eabi-gcc --version
arm-none-eabi-gcc (GNU Tools for ARM Embedded Processors) 5.4.1 20160919 (release) [ARM/embedded-5-branch revision 240496]
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

C:\Users\User>python --version
Python 2.7.13

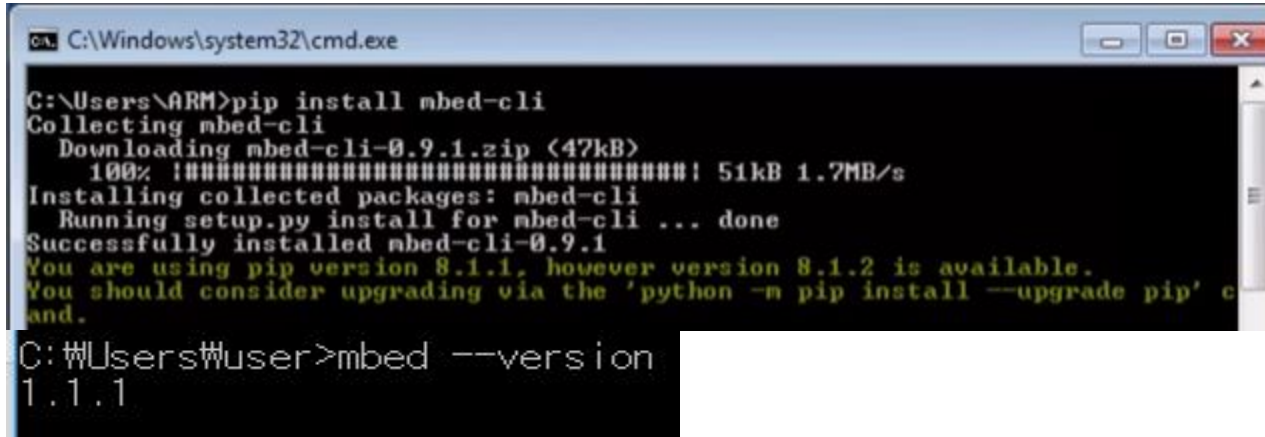
C:\Users\User>pip --version
pip 9.0.1 from c:\python27\lib\site-packages (python 2.7)

C:\Users\User>hg --version
Mercurial Distributed SCM (version 4.2+4)
(see https://mercurial-scm.org for more information)

Copyright (C) 2005-2017 Matt Mackall and others
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

# mbed CLI prerequisite

Step 2 install mbed CLI from pip



```
C:\Windows\system32\cmd.exe

C:\Users\ARM>pip install mbed-cli
Collecting mbed-cli
  Downloading mbed-cli-0.9.1.zip (47kB)
    100% |#####| 51kB 1.7MB/s
Installing collected packages: mbed-cli
  Running setup.py install for mbed-cli ... done
Successfully installed mbed-cli-0.9.1
You are using pip version 8.1.1, however version 8.1.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' c
and.

C:\Users\user>mbed --version
1.1.1
```

# mbed Quick Start

## Step 1 Import

: copy URL of

repository([https://developer.mbed.org/mbed\\_demo/code/mbed\\_blinky](https://developer.mbed.org/mbed_demo/code/mbed_blinky))

```
E:\Heedong\Work\mbed\Src\dev_directory>mbd import https://developer.mbed.org/users/mbed_demo/code/mbed_blinky/
[mbd] Importing program "mbed_blinky" from "https://developer.mbed.org/users/mbed_demo/code/mbed_blinky" at latest revision in the current branch
[mbd] Adding library "mbed" from "http://mbed.org/users/mbed_official/code/mbed/builds" at rev #f9eeca106725
[mbd] Downloading library build "f9eeca106725" (might take a minute)
[mbd] Unpacking library build "f9eeca106725" in "E:\Heedong\Work\mbed\Src\dev_directory\mbed_blinky\mbed"
[mbd] Couldn't find build tools in your program. Downloading the mbed 2.0 SDK tools...
[mbd] Auto-installing missing Python modules...
```

# mbed Quick Start

## Step 2 toolchain and target

- 1) Go to program folder
- 2) Detect toolchain and target options after connecting the dev board to the PC via USB
- 3) compile

```
C:\Users\ARM\mbed_blinky>mbed compile
Building project mbed_blinky (K64F, GCC_ARM)
Scan: .
Scan: mbed
Scan: env
Compile: main.cpp
Link: mbed_blinky
Elf2Bin: mbed_blinky

+-----+-----+-----+
| Module | .text | .data | .bss |
+-----+-----+-----+
| Fill   | 81    | 4     | 2473 |
| Misc   | 35190 | 2228  | 423  |
| Subtotals | 35271 | 2232  | 2896 |
+-----+-----+-----+

Allocated Heap: 65536 bytes
Allocated Stack: 32768 bytes
Total Static RAM memory (data + bss): 5128 bytes
Total RAM memory (data + bss + heap + stack): 103432 bytes
Total Flash memory (text + data + misc): 38543 bytes
Image: .\build\K64F\GCC_ARM\mbed_blinky.bin
```

```
C:\Users\ARM\mbed_blinky>mbed detect

[mbed] Detected K64F, port None, mounted F:
[mbed] Supported toolchains for K64F

+-----+-----+-----+-----+-----+-----+
| Target | mbed OS 2 | mbed OS 5 | ARM   | GCC_ARM | uARM   | GCC_CR |
+-----+-----+-----+-----+-----+-----+
| K64F   | -         | -         | Supported | Supported | -      | -      |
+-----+-----+-----+-----+-----+-----+

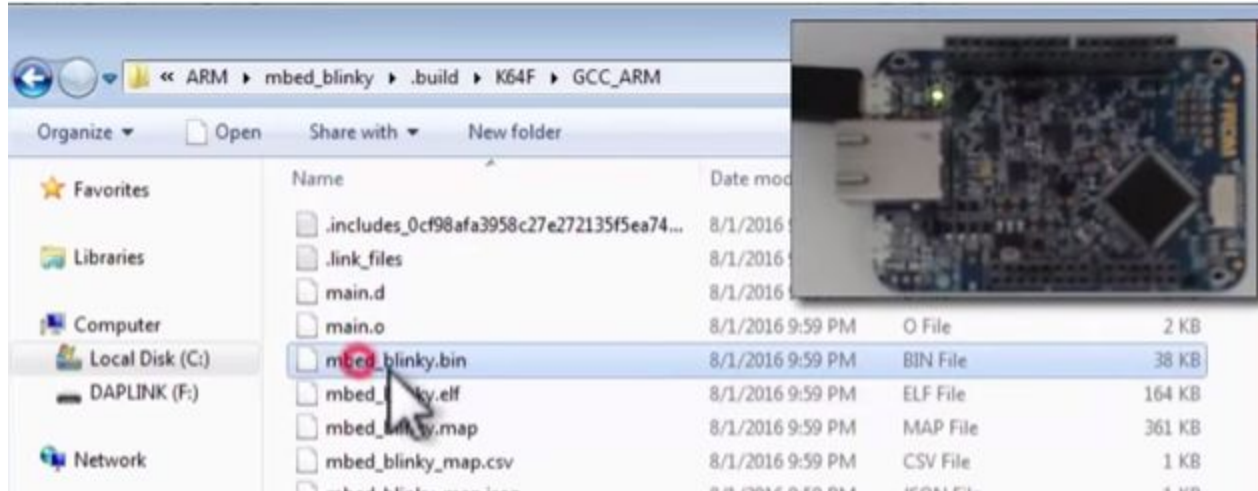
Supported targets: 1
Supported toolchains: 3

C:\Users\ARM\mbed_blinky>mbed toolchain GCC_ARM
[mbed] GCC_ARM now set as default toolchain in program "mbed_blinky"

C:\Users\ARM\mbed_blinky>mbed target K64F
[mbed] K64F now set as default target in program "mbed_blinky"
```

# mbed Quick Start

Step 3 Find .bin in .build directory and copy binary(.bin) to board



Step 4 Reset board to run the code

Step 5 Modify code in main.cpp



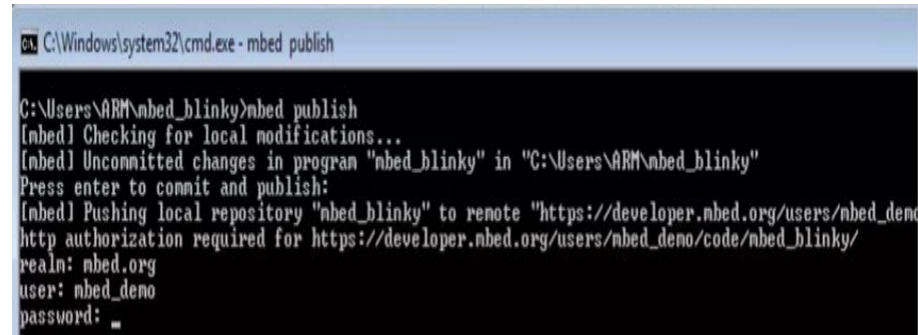
# mbed Quick Start

Step 6 Change blink speed and LED colors and Compile and Flash new binary to Board

Step 7 publish

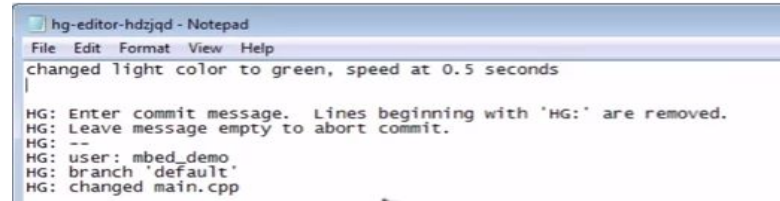
- publish code back to server and press enter and fill in change message
- close window to continue
- enter mbed username and mbed password
- see changes on website (History tab)

Step 8 Reset board to run the code



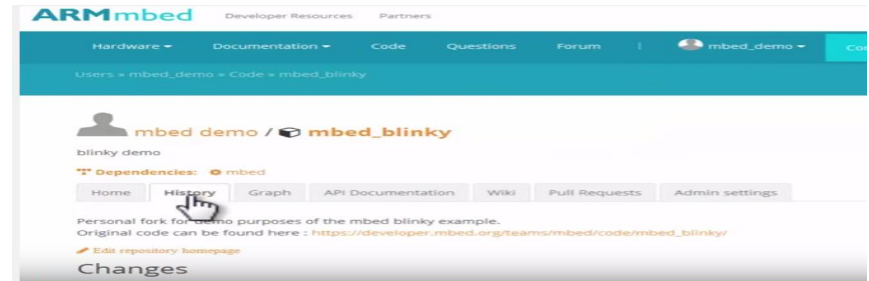
```
C:\Windows\system32\cmd.exe - mbed publish

C:\Users\ARM\mbed_blinky>mbed publish
[mbed] Checking for local modifications...
[mbed] Uncommitted changes in program "mbed_blinky" in "C:\Users\ARM\mbed_blinky"
Press enter to commit and publish:
[mbed] Pushing local repository "mbed_blinky" to remote "https://developer.mbed.org/users/mbed_demo"
http authorization required for https://developer.mbed.org/users/mbed_demo/code/mbed_blinky/
realm: mbed.org
user: mbed_demo
password: 
```



```
hg-editor-hdzjqd - Notepad
File Edit Format View Help
changed light color to green, speed at 0.5 seconds

HG: Enter commit message. Lines beginning with 'HG:' are removed.
HG: Leave message empty to abort commit.
HG: --
HG: user: mbed_demo
HG: branch 'default'
HG: changed main.cpp
```



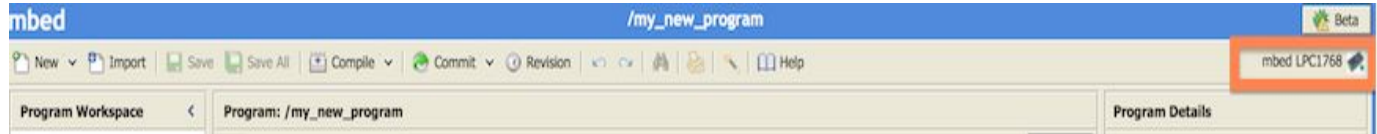
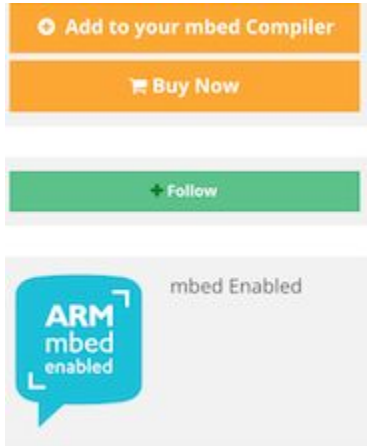
# How to compile mbed platform

mbed online compiler

# Importing blinky

Selecting a target board and Adding a board to your list

Go to the board's page on mbed.com(<https://developer.mbed.org/platforms/>) and click the Add to your mbed compiler button



# Importing blinky

To get blinky into the mbed online compiler, click the Import into mbed IDE button

 **main.cpp**  


```
#include "mbed.h"


DigitalOut led1(LED1);

// main() runs in its own thread in the OS
int main() {
    while (true) {
        led1 = !led1;
        wait(0.5);
    }
}
```

**Import Program**

Import a program from mbed.org into your workspace.



 Please specify name

**Source URL:**

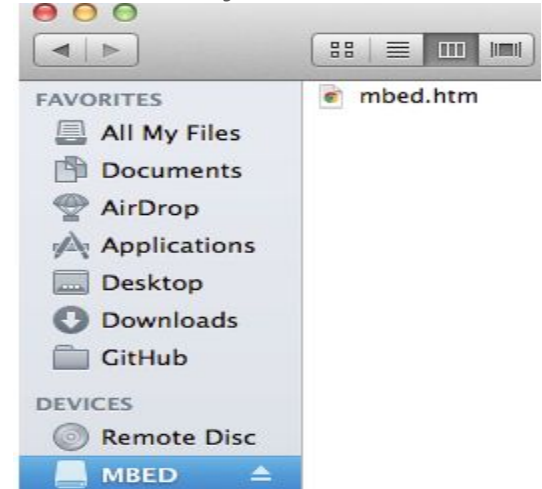
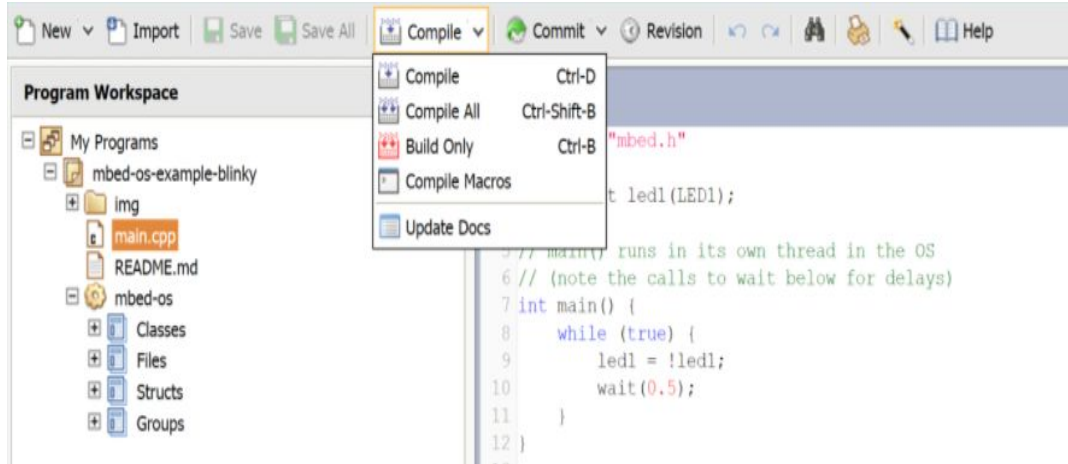
**Import As:** ☒ Program ☐ Library

**Import Name:**

**Update:** ☐ Update all libraries to the latest revision

# Compile and install

The mbed online compiler builds a .bin file that you can install on your board

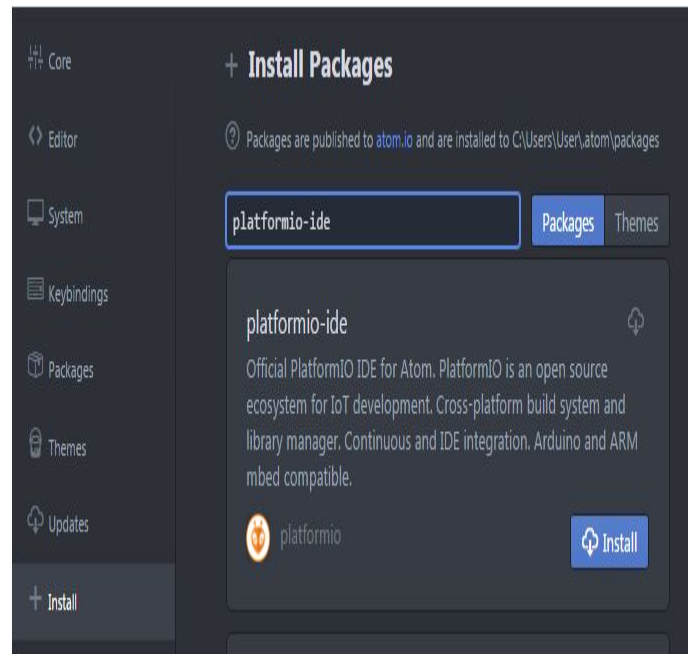
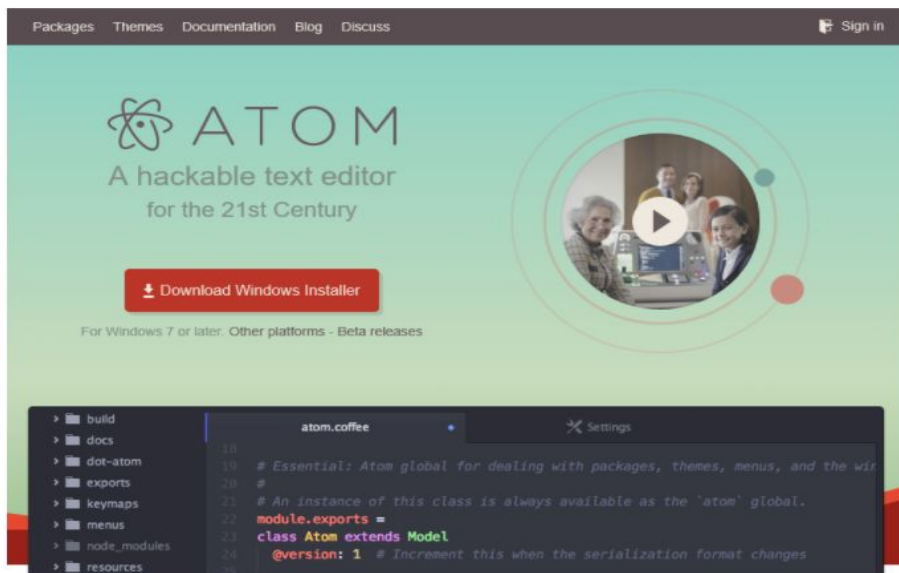


Connect your board to your computer over USB. mbed boards are shown as “removable storage”

Drag and drop your program to the board / The board installs the program and Reset the board and see the LED blink

# How to install platform io

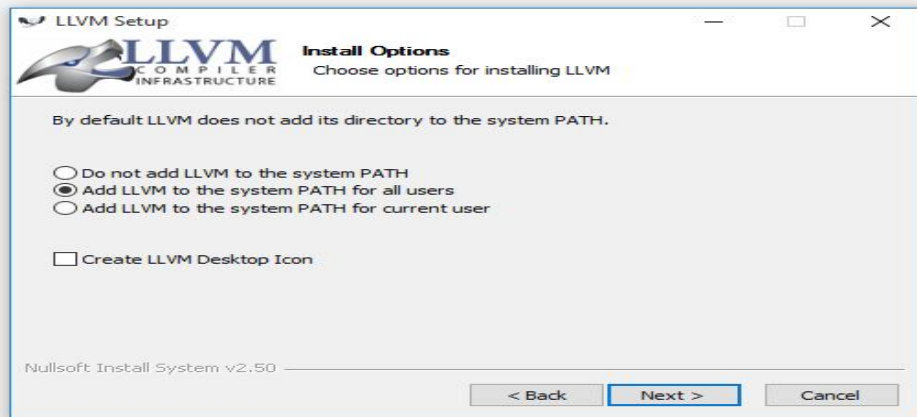
## 1. Install atom package



# How to install platform io

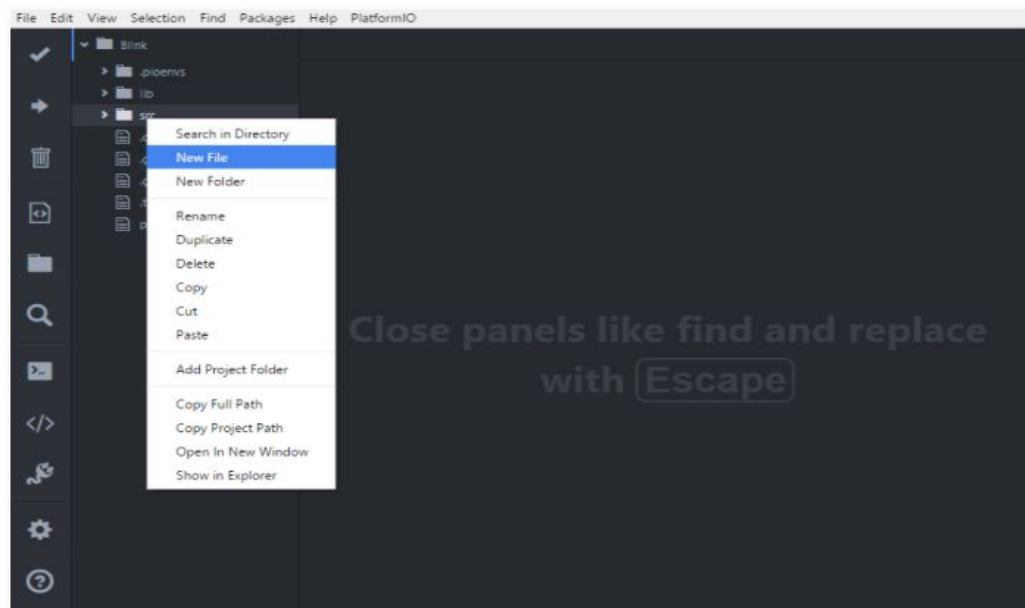
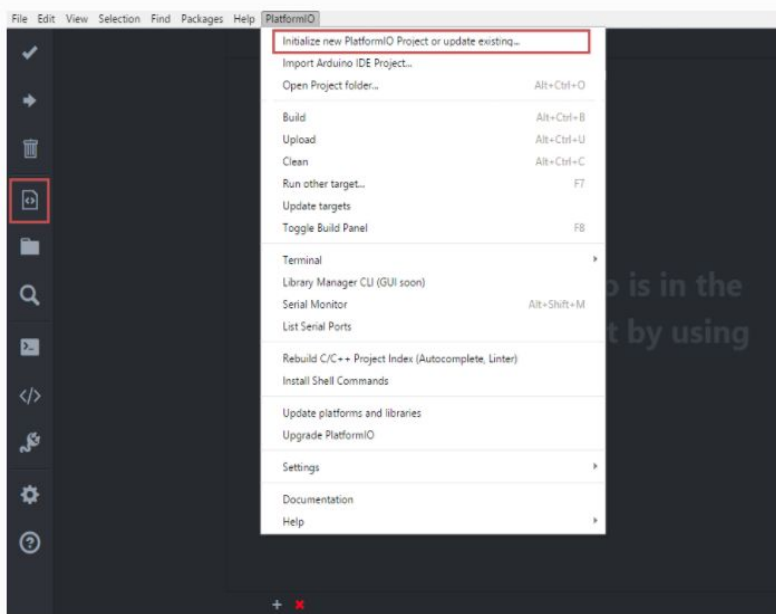
## 2. Clang for intelligent code completion

- **Mac OS X:** Install the latest Xcode along with the latest Command Line Tools (they are installed automatically when you run `clang` in Terminal for the first time, or manually by running `xcode-select --install`)
- **Windows:** Download Clang 3.9.1 for Windows. Please select "Add LLVM to the system PATH" option on the installation step.
  - Clang 3.9.1 for Windows (32-bit)
  - Clang 3.9.1 for Windows (64-bit)



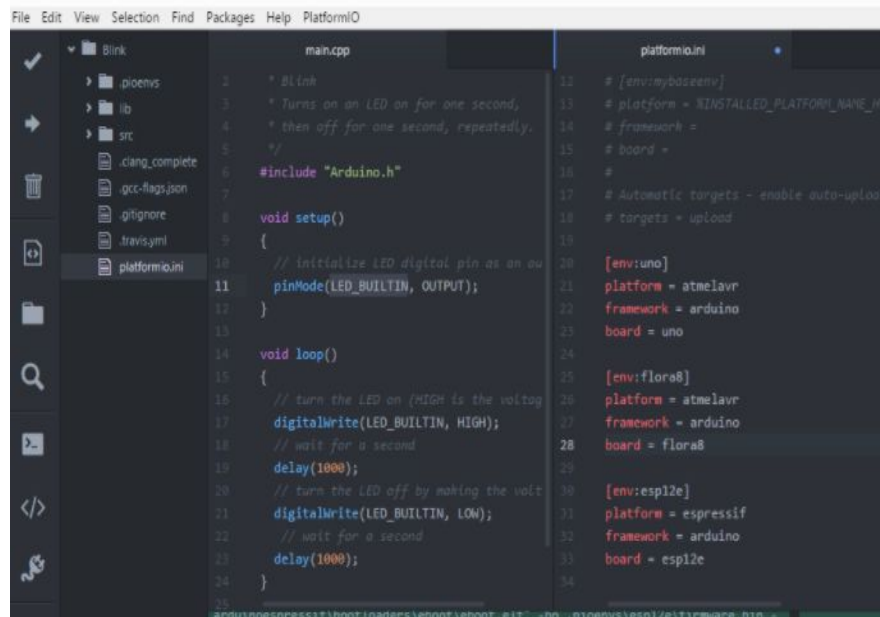
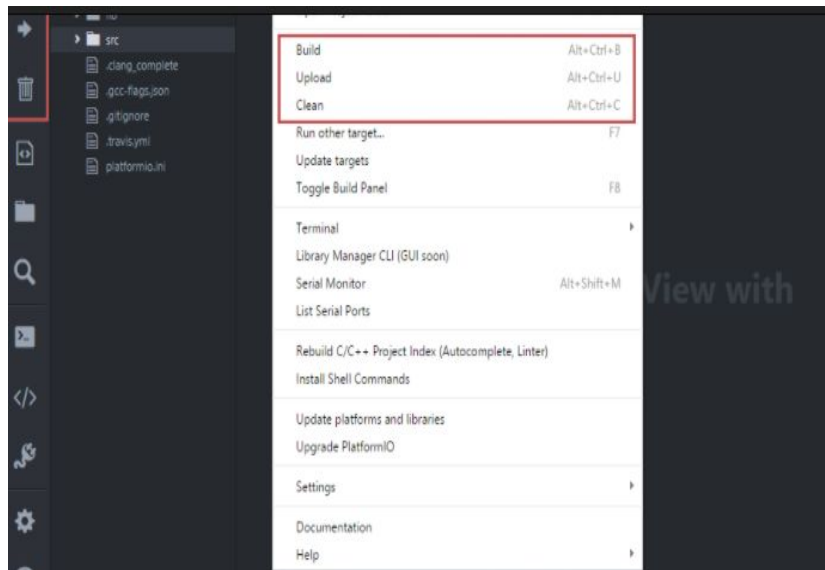
# How to build platform io

## Set up new project

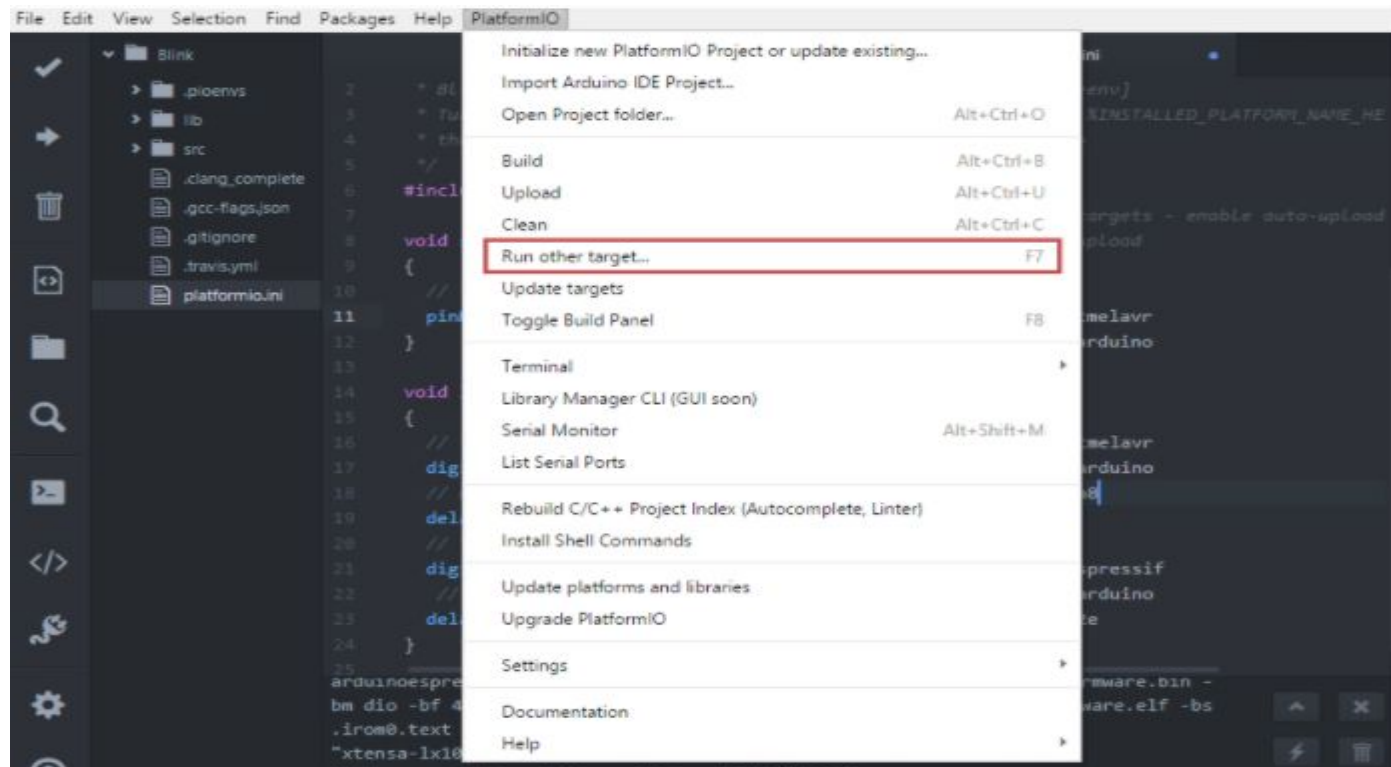




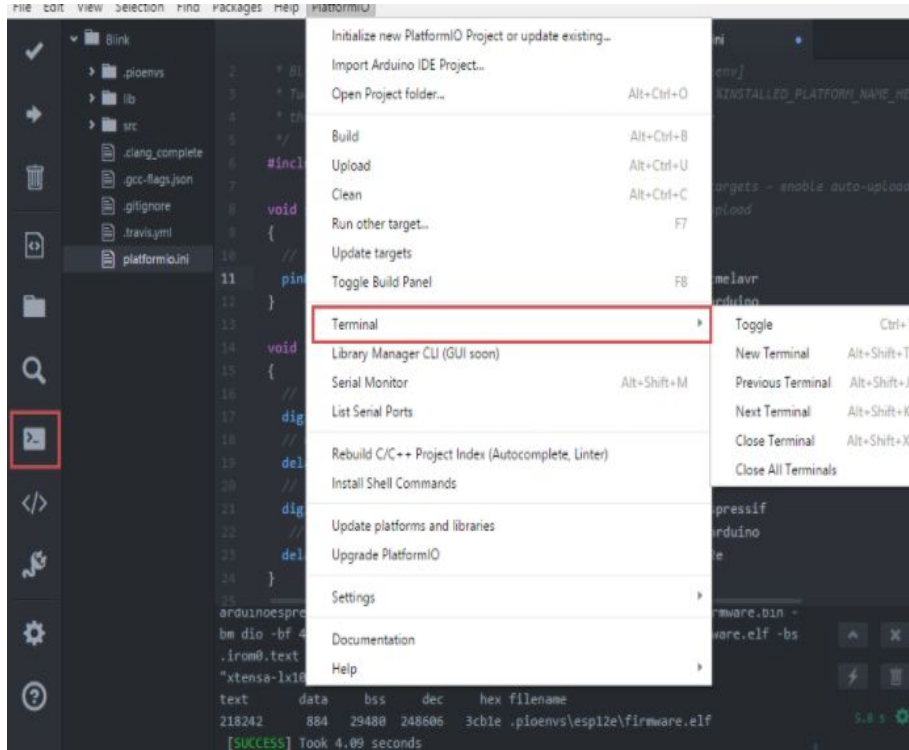
# How to build platform io



# How to run platform io



# How to install platform io

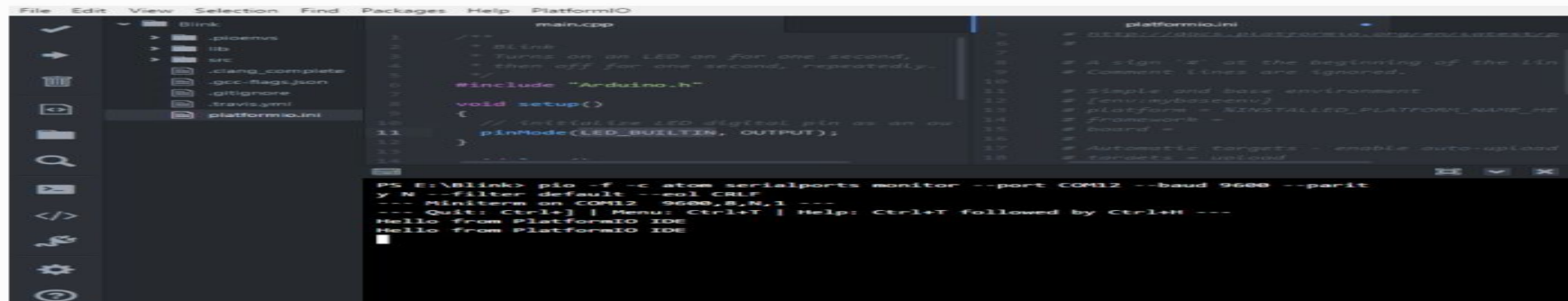
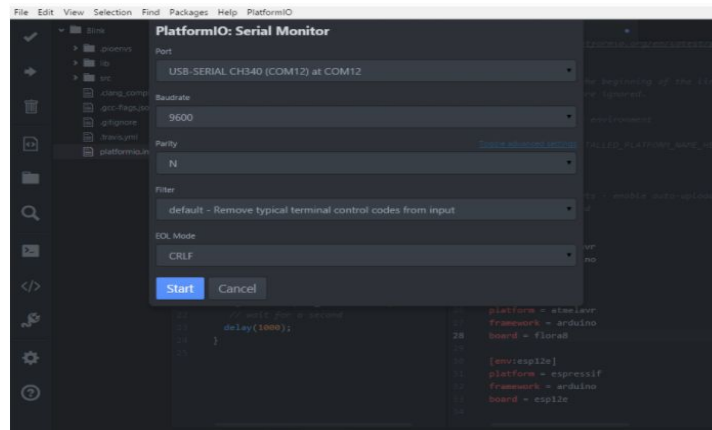
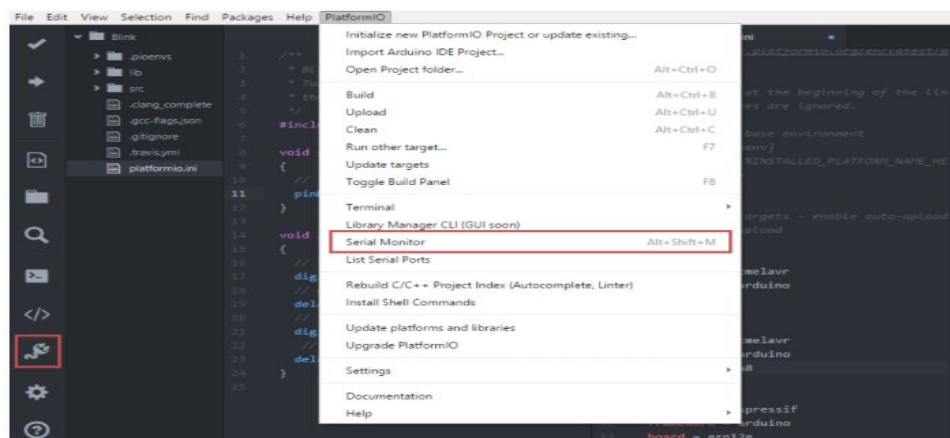


```
PS E:\Blink> pio --help
Usage: pio [OPTIONS] COMMAND [ARGS]...

Options:
  --version          Show the version and exit.
  -f, --force        Force to accept any confirmation prompts.
  -c, --caller TEXT  Caller ID (service).
  -h, --help         Show this message and exit.

Commands:
  boards      Pre-configured Embedded Boards
  ci          Continuous Integration
  init        Initialize new PlatformIO based project
  lib         Library Manager
  platforms   Platforms and Packages Manager
  run         Process project environments
```

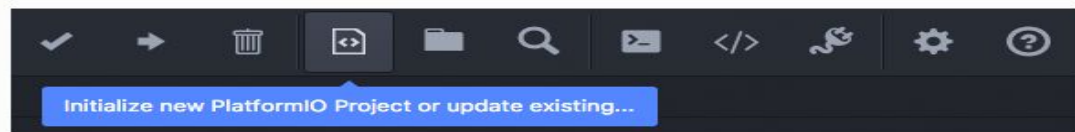
# How to install platform io



# Platform IO Toolbar

## PlatformIO Toolbar

PlatformIO IDE Toolbar contains quick access buttons for the popular commands. Each button contains hint (delay mouse on it).



- PlatformIO: Build
- PlatformIO: Upload
- PlatformIO: Clean
- ||
- Initialize new PlatformIO Project or update existing...
- Add/Open Project Folder...
- Find in Project...
- ||
- Terminal
- Library Manager
- Serial Monitor
- ||
- Settings
- PlatformIO Documentation

# How to build platform IO CLI

Install shell commands on platform io

## How to build PlatformIO based project

---

1. Install PlatformIO Core
2. Download [examples source code](#)
3. Extract ZIP archive
4. Run these commands:

```
# Change directory to example
> cd platformio-examples/mbed/mbed-blink

# Build project
> platformio run

# Upload firmware
> platformio run --target upload

# Build specific environment
> platformio run -e lpc1768

# Upload firmware for the specific environment
> platformio run -e lpc1768 --target upload

# Clean build files
> platformio run --target clean
```