

Agora | Getting started with Pelion on Agora

3 Dec 2021 | v1.0.0



www.embeddedplanet.com



1. **Install toolchain and mbed-cli**

Follow the instructions here to install mbed-cli and the GCC\_ARM toolchain for Windows or MacOS:

<https://os.mbed.com/docs/mbed-os/v6.1/quick-start/build-with-mbed-cli.html>

Follow these instructions for installing mbed-cli and GCC\_ARM on Linux:

1. Check if python is already installed on your system by running python --version in your terminal. If "Python x.x.x" prints out, then you already have python and you can skip to step 3.
2. Install Python from their site (<https://www.python.org/downloads/>), or search the web for instructions to install Python for your distribution.
3. Run pip install virtualenv. If the installation fails with a "Permission denied" error, run pip install --user virtualenv instead.
4. Make a directory for this example:

mkdir ~/agoraTest

**cd** ~/agoraTest

1. Create a python virtual environment: python -m virtualenv venv If you have multiple python interpreters, then try running the virtualenv command, specifying the python executable with -p
2. Source the virtual environment: source venv/bin/activate. This will need to be run again if you exit the terminal.
3. Now install the mbed-cli by running pip install mbed-cli mercurial

For the toolchain installation, navigate to <https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads> and click gcc-arm-none-eabi-9-2020-q2-update-x86\_64-linux.tar.bz2.

Now, you need to extract the toolchain and tell mbed-cli where to find it:

1. Make a toolchains directory: mkdir ~/toolchains
2. Copy the download to that directory. (ex. cp ~/Downloads/gcc-arm-none-eabi-9-2020-q2-update-x86\_64-linux.tar.bz2 ~/toolchains).
3. Then, cd toolchains.
4. Lastly tar xf gcc-arm-none-eabi-9-2020-q2-update-x86\_64-linux.tar.bz2.
5. Now run mbed config -G GCC\_ARM\_PATH ~/toolchains/gcc-arm-none-eabi-9-2020-q2-update/bin. This globally tells mbed-cli where to find the toolchain. Every project you make that uses this mbed-cli will also use the toolchain you configured here, assuming you use the mbed-cli installation from this python virtual environment.

Now, you're ready to clone the example cloud client. Go back to your projects directory (cd ~/agoraTest) and run mbed import https://github.com/embeddedplanet/mbed-os-example-pelion. In addition to fetching the mbed-os-example-pelion source code, it recursively fetches its dependencies such as ep-oc-mcu, which allows the MCU to interface with the sensors the Agora offers.

1. **IDE setup (optional)**

Go to the root source directory (cd ~/agoraTest/mbed-os-example-pelion).

Run mbed export -i <Your IDE> -m EP\_AGORA, replacing <Your IDE> with one of the following:

cces, cmake\_gcc\_arm, codeblocks, e2studio,

eclipse\_armc5, eclipse\_gcc\_arm, eclipse\_iar, embitz,

gcc\_arm, gnuarmeclipse, iar, make\_armc5,

make\_armc6, make\_gcc\_arm, make\_iar, mcuxpresso,

netbeans, qtcreator, sw4stm32, uvision,

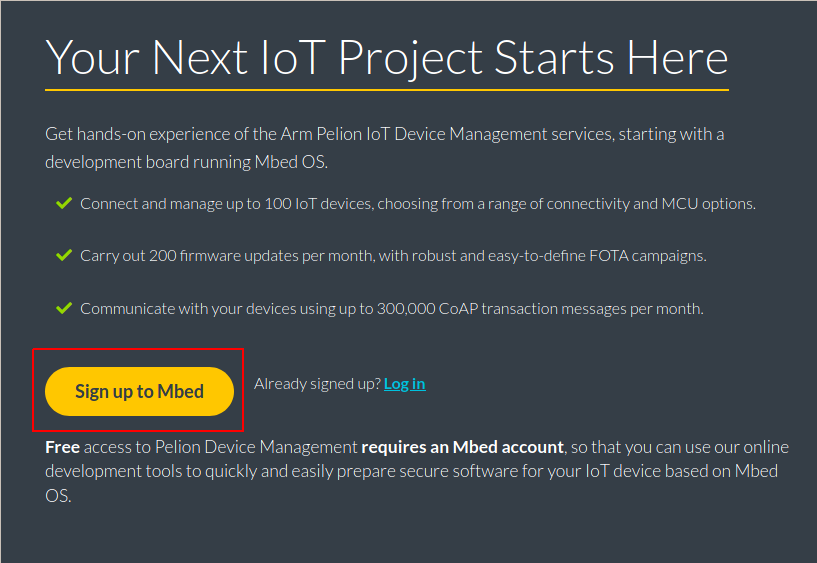
uvision5, uvision6, vscode\_armc5, vscode\_gcc\_arm,

vscode\_iar

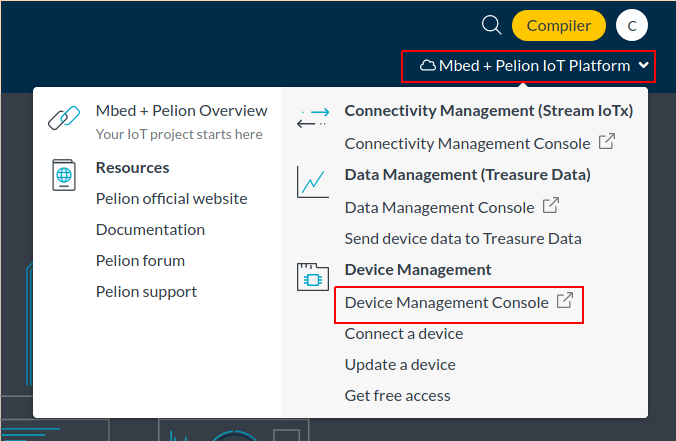
Now you can open the project in your IDE of choice, and you can compile the project similarly to how you would compile other projects in your IDE.

For more information, like making adjustments so you can use a debugger, check out: <https://os.mbed.com/docs/mbed-os/v6.2/debug-test/third-party-tools.html>

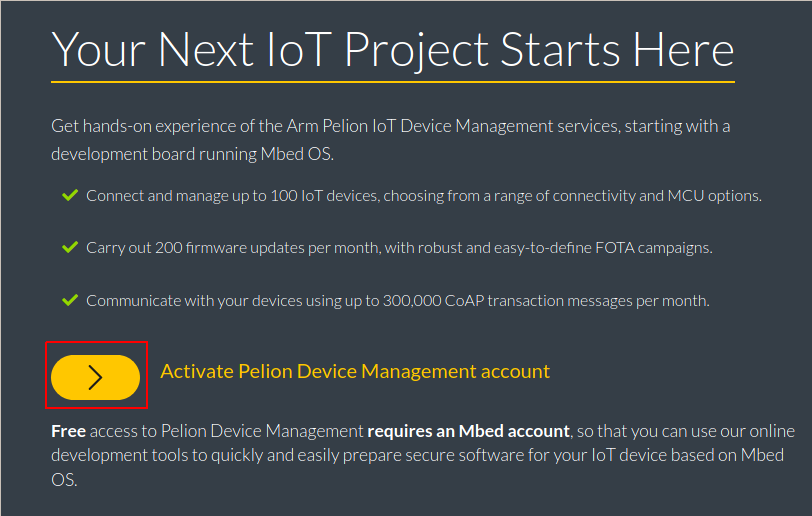
1. **Pelion Account and API Key Setup**
2. In your browser, navigate to <https://os.mbed.com/pelion-free-tier/>: Click on "Sign up to Mbed" and follow the prompts to create a new account if you don't have an account with Mbed already.



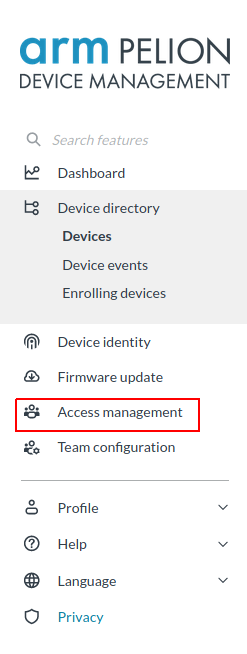
1. Answer Mbed's prompts, then confirm your email by clicking the link in the email that was sent to you when you created your account.
2. If the next button At the top right corner of the page, hover over the "Mbed + Pelion IoT Platform" and then click on "Device Management Console."



1. In the new tab, click on "Log in with account.mbed.com" and then "Click here to register." After being redirected back to the "Your Next IoT Project Starts Here" screen, click on "Activate Pelion Device Management account."



1. Then, hover over "Mbed + Pelion IoT Platform" and open "Device Management Console." Agree to the terms and conditions.
2. On the left panel, click on "Access management," then click "API keys" in the submenu.



1. Click on "New API key" in the top right corner, then type in a suitable name for your new key. Click on the Group dropdown and choose "Developers." Then, click "Create API Key." **Copy this key and save it somewhere so you don't lose it!**This key only appears once. If you accidentally lose the key, you will need to create a new API key again.
2. **Building the Client Source Code with the API Key**

Use the terminal to tell the mbed build system about the new API key you made by running the following within the mbed-os-example-pelion directory:

mbed config -G CLOUD\_SDK\_API\_KEY <API Key> Replace <API Key> with the key you saved earlier. The -G makes mbed-cli reuse the same API Key for every new project. Again, this assumes you're using the mbed-cli installation in this python virtual environment.

Tell mbed-cli to generate an update certificate, a private key, and other developer cloud credentials: mbed device-management init -d arm.com --model-name example-app --force -q The credentials here are *NOT* suited for production. They are only meant for testing update campaigns on devices designated for development.

Run the following to tell the mbed build system about the target and toolchain:

mbed target EP\_AGORA

mbed toolchain GCC\_ARM

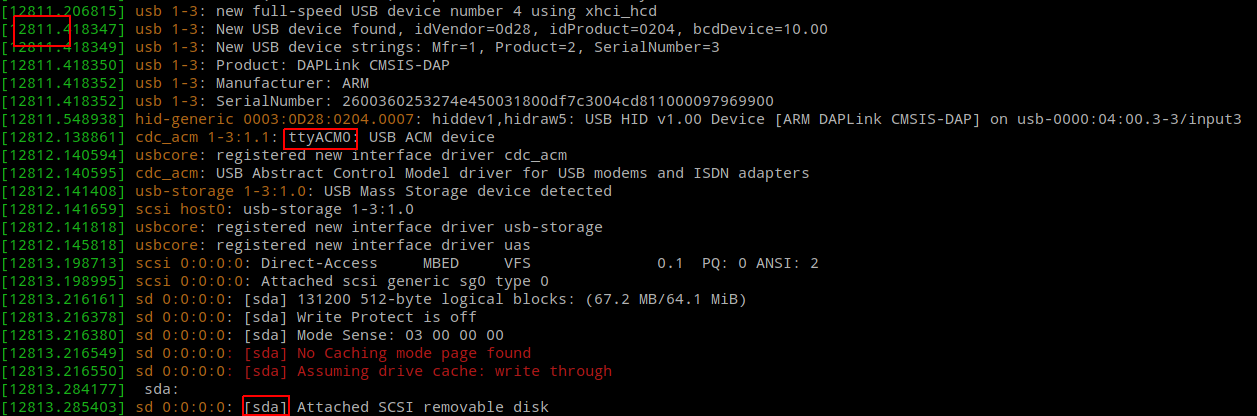
Now, to build the code:

mbed compile

Note: Alternatively, target config and compilation can be consolidated into one line:

mbed compile -m EP\_AGORA -t GCC\_ARM

1. **Upload to the Board**
2. Plug the Agora into the Flidor, then plug the usb micro into DHD Debug (NOT Target USB!). Plug the other end into your PC.
3. The DAPLink firmware on the Flidor will enumerate as 3 different USB devices, which are mass storage, COM/CDC, and human interface. These exist for code uploading, UART output from the target board, and debugging, respectively.
4. If you're using Ubuntu, then the mass storage device should automatically mount as the DAPLINK directory in /media/<your username>/DAPLink. If it doesn't automatically mount, you may need to figure out which device it enumerates as, then mount it manually (For manual mounting, run dmesg shortly after plugging the Flidor in and find the line with "Attached SCSI removable disk." The letters in the brackets to the left represent the block device name). The line containing "USB ACM device" will also show the enumeration of the CDC device, which will be useful shortly.



1. Then, copy the build output (~/agoraTest/mbed-os-example-pelion/BUILD/EP\_AGORA/GCC\_ARM/mbed-os-example-pelion.hex) to DAPLINK. You should see a green LED blinking on the Flidor. If you see the file in the DAPLink directory and the green light isn't blinking, try running sync in the terminal. The sync command tells Linux to finish all file transfers immediately rather than caching them. If that doesn't work, unplug the flidor from your PC then try uploading again.
2. **View Debug Output**

Open a Serial Monitor. Minicom will be used as an example here, but you can install PuTTY, TeraTerm, etc. :

sudo minicom -D <CDC Device> -b 115200

In this example, the Flidor enumerated as ttyACM0, so <CDC Device> would be replaced with /dev/ttyACM0.

It may also be beneficial to check if flow control is off. In minicom, the setting can be found by typing Ctrl-A then o, then Configure Serial Port. Lastly, press f to toggle hardware flow control.

Then press the "Target Reset" button on the Flidor to see the serial output from boot. You should be seeing something like the following:

Mbed Bootloader

No Update image

[DBG ] Active firmware up-to-date

booting...

Application ready

Connect **to** network

Network initialized, connected with IP x.x.x.x

Start developer flow

...

At this point, the device should start registering with Pelion.

From the menu on the left in the Pelion Device Management Portal, select "Device Directory" and then "Devices."

The device should be showing in the list. To show the device resources, click on the device ID in the list, then select "Resources." Now, scroll down to select a sensor reading and click on the "Path" of the reading. Note that some values may have more than one reading because more than one sensor could be providing such reading. Here is how to obtain a temperature reading for example:

Temperature Reading