

Arduino CLI on Windows 10 Enterprise

Step-by-step Installation and Verification Guide

1. Download the Arduino CLI binary

1. Open a web browser and go to:
<https://github.com/arduino/arduino-cli/releases>
2. In the Releases page, find the latest version.
3. Under Assets for that version, download the file named similar to:
arduino-cli_version_Windows_64bit.zip
(The exact version number will vary, e.g. arduino-cli_1.0.4_Windows_64bit.zip)
You might need to scroll down and choose 'show all assets'

2. Extract the executable

1. Locate the downloaded .zip file in your Downloads folder.
2. Right-click the .zip file and choose "Extract All...".
3. Choose a folder for the CLI, for example:
C:\Tools\ArduinoCLI
4. After extraction, you should see a file named:
arduino-cli.exe
inside that folder.

3. Add Arduino CLI to the PATH (so you can call it from any terminal)

1. Press Windows key, type:
environment variables
and choose "Edit the system environment variables".
2. In the System Properties window, click "Environment Variables...".
3. Under "User variables" (for your account), select the "Path" entry and click "Edit...".
4. Click "New" and paste the folder where arduino-cli.exe is located, e.g.:
C:\Tools\ArduinoCLI
5. Click "OK" on all dialogs to close them.

4. Open a terminal and verify the installation

1. Press Windows key, type:
cmd
and open "Command Prompt".
2. In the terminal, type:
arduino-cli version
3. You should see output similar to:
arduino-cli Version: x.y.z
If you get "'arduino-cli' is not recognized as an internal or external command",

check that the PATH entry points to the correct folder and reopen the terminal.

5. Initialize the Arduino CLI configuration (first-time setup)

1. In Command Prompt, run:

```
arduino-cli config init
```

2. This creates the default configuration file in your user directory.

3. To confirm, you can run:

```
arduino-cli config dump
```

and you should see the current configuration printed.

6. Update the core index (list of supported boards)

1. In Command Prompt, run:

```
arduino-cli core update-index
```

2. Wait until it completes successfully. This downloads the latest board metadata.

7. Verify that the CLI can see your Arduino cores

1. In Command Prompt, run:

```
arduino-cli core list
```

2. You should see at least:

arduino:avr (not installed or installed, depending on your setup)

3. To install the official AVR core (for boards like Arduino Uno), run:

```
arduino-cli core install arduino:avr
```

4. After installation, re-run:

```
arduino-cli core list
```

and confirm that arduino:avr shows as "installed".

8. Install the Adafruit AVR core (for boards like ItsyBitsy 32u4)

1. Add the Adafruit board manager URL to your config:

```
arduino-cli config add board_manager.additional_urls
```

```
https://adafruit.github.io/arduino-board-index/package_adafruit_index.json
```

2. Update the core index again:

```
arduino-cli core update-index
```

3. Install the Adafruit AVR core:

```
arduino-cli core install adafruit:avr
```

4. Accept USB driver installation and then Verify:

```
arduino-cli core list
```

and check that adafruit:avr appears as installed.

9. Test that the CLI can see your Arduino board

1. Connect your Arduino (or compatible) board via USB to the Windows 10 machine.

2. In Command Prompt, run:

arduino-cli board list

3. You should see an entry with:

- A COM port, e.g. COM3
- A "Type" (e.g. Serial Port (USB))
- A board name and FQBN if the core is installed.

10. Test a simple compile (no upload yet)

1. Create a folder for sketches, e.g.:

C:\Users\<YourUser>\Arduino\BlinkTest

2. Inside that folder, create a file named BlinkTest.ino with a minimal example, for

example:

```
void setup() {  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(LED_BUILTIN, HIGH);  
    delay(500);  
    digitalWrite(LED_BUILTIN, LOW);  
    delay(500);  
}
```

3. In Command Prompt, navigate to the sketch folder:

cd C:\Users\<YourUser>\Arduino\BlinkTest

4. Compile the sketch for a known board (example: Arduino Uno):

arduino-cli compile --fqbn arduino:avr:uno .

5. If the compile finishes without errors, your CLI toolchain is working correctly.

11. Optional: Test upload (if you have a supported board connected)

1. Identify the COM port from "arduino-cli board list" (for example, COM3).

2. Upload the compiled Blink sketch (example for Arduino Uno):

arduino-cli upload -p COM3 --fqbn arduino:avr:uno .

3. If the sketch uploads successfully and the on-board LED begins blinking, your

Arduino CLI installation and workflow are fully verified.

Summary

- You downloaded the arduino-cli Windows binary.
- Added it to your PATH so it can be run from any terminal.
- Verified with "arduino-cli version".
- Initialized configuration and updated board core index.
- Installed at least one board core (e.g., arduino:avr).
- Successfully compiled (and possibly uploaded) a test sketch.

This completes the basic Arduino CLI setup and verification on Windows 10 Enterprise.