Upgrading from v1.0 to v2.0

EV3 MicroPython version 2.0 was released on May 18, 2020.

This section is for users who have previously used LEGO MINDSTORMS EV3 MicroPython v1.0. We'll explain what's changed and how you can upgrade to benefit from the latest improvements.

If you are a new user and you just got started using version 2.0, you may skip this section.

Upgrading the microSD Card

To upgrade, download the latest microSD card file and install it using the standard instructions.

Note that this will erase all your existing files on the SD Card. Before you upgrade, make sure that you still have all your projects on your computer. If not, you can upload files back to your computer using these instructions.

As with any software update, *be careful about when you update*. For example, if you developed your code using version v1.0 and you are halfway into your robotics competition season, you may want to stick with v1.0 for now.

Upgrading your existing programs

Most changes in v2.0 are *new* features, like support for additional sensors. Naturally, this will not affect your existing code. However, some changes were made to existing features to improve performance.

All originally documented features in v1.0 will still work after you upgrade. This means that most programs originally made for v1.0 will work with the v2.0 microSD card image without any changes.

To try this, simply download and run your original code as you did before.

However, it is recommended that you upgrade both the microSD card and your programs at the same time to ensure everything works as expected.

The new EV3Brick() class replaces the ev3brick module

Version 2.0 introduces the `EV3Brick()` class. You can use it instead of the old `ev3brick` module. The old `ev3brick` module can still be used, but it is no longer recommended or documented.

The `EV3Brick()` class improves the speed and reliability of the EV3 screen and the EV3 speaker. It also adds functionality like speech and drawing shapes. The default font size is also bigger to make it easier to read text on the screen.

You can use the following table as a starting point to upgrade your scripts. See the `EV3Brick()` class documentation for complete details of all methods and arguments.

v1.0	v2.0

`from pybricks import ev3brick as brick`	`from pybricks.hubs import EV3Brick`
	`ev3 = EV3Brick()`

`brick.light(Color.RED)`	`ev3.light.on(Color.RED)`
`brick.light(None)`	`ev3.light.off()`

`if Button.LEFT in brick.buttons():`
`    print("Left is pressed.")`
	`if Button.LEFT in ev3.buttons.pressed():`
	`    print("Left is pressed.")`

`brick.sound.beep()`	`ev3.speaker.beep()`
`brick.sound.file(SoundFile.HELLO)`	`ev3.speaker.play_file(SoundFile.HELLO)`
	`ev3.speaker.say("I can say anything!")`
	`ev3.speaker.play_notes(['C4/4', 'G4/4'])`
`brick.display.text("Hello!", (50, 60))`	`ev3.screen.draw_text(50, 60, "Hello!")`

`brick.display.text("Hello")`
`brick.display.text("world!")`
	`ev3.screen.print("Hello")`
	`ev3.screen.print("world!")`

	`from pybricks.media.ev3dev import Font`
	
	`big_font = Font(size=24)`
	`ev3.screen.set_font(big_font)`

`from pybricks.parameters import ImageFile`
 	
`brick.display.image(ImageFile.QUESTION_MARK)`
	`from pybricks.media.ev3dev import ImageFile`
	
	`ev3.screen.load_image(ImageFile.Question_MAR`

	`ev3.screen.draw_line(30, 30, 30, 100)`
	`ev3.screen.draw_box(50, 30, 90, 60)`
	`ev3.screen.draw_circle(70, 90, 20, fill=True`

`brick.battery.voltage()`	`ev3.battery.voltage()`

Other internal changes to existing features

- Most methods of the `Motor()` class now have `Stop.HOLD` as the default instead of `Stop.COAST`. This improves accuracy in most applications. You can still select `Stop.COAST` if you like.
- The internal PID controllers for the motors are more accurate than before. If you give a motor command when it is already running, it smoothly adjusts the speed to the newly given command. This works even if you keep adjusting the speed in a fast loop.
- Methods to configure motor settings have changed. You can change settings using the control attribute now. The old settings setters continue to exist in the implementation, but they are no longer documented.
- So-called Python *keyword arguments* are now supported. Previously, you could only enter the argument *values*. For example:

  ```
  my_motor.run_angle(500, 90, Stop.HOLD, False)
  ```

  This is still possible. But you can now choose to omit optional arguments and specify others with *keywords*. This can make your code easier to read. For example:

  ```
  my_motor.run_angle(500, 90, wait=False)
  ```

- It is no longer necessary to import `pybricks.tools.print`. The `print` function is now built-in. It works just like Python or MicroPython.

- Most parameters in the `parameters` now have a specific type and representation. For example, suppose you measure a color and print the result. If you do `print(Color.RED)`, you will see the parameter instead of a technical number.
- Sound and image files have moved to a dedicated media module. Importing them from the old location will continue to work in this release, to make sure existing scripts will still work.

Installing an older version of the Visual Studio Code extension

The Visual Studio Code extension and this documentation are updated automatically. You can still use your existing scripts with the updated extension. If you absolutely wish to keep the old version, look for the EV3 extension on the extension tab, click the gear icon, and click *install another version*.