Getting started with LEGO® MINDSTORMS® Education EV3 MicroPython

Version 2.0.0

TABLE OF CONTENTS

This guide shows you how to get started writing MicroPython programs for your LEGO® MINDSTORMS® EV3 robots. You'll learn to do so in two steps:

- Installation: First you'll prepare your computer and your EV3 Brick by collecting and installing the required tools. You'll also learn how to turn the EV3 Brick on and off and to navigate the menu on the screen.
- Creating and running programs: Next, you'll learn how to create a program and download it to the EV3 Brick. You'll also learn how to start that program from your computer or from the EV3 Brick.

After you've run the first demo program, you'll be ready to try out the example programs and start inventing your own programs.

CHAPTER ONE

INSTALLATION

This page guides you through the steps to collect and install everything you need to start programming.

1.1 What do you need?

To get started, you'll need:

- A Windows 10 or Mac OS computer
- Internet access and administrator access

  This is required during the installation only. You will not need special access to write and run programs later on.
- A microSD card

  You'll need a card with a minimum capacity of 4GB and a maximum capacity of 32GB. This type of microSD card is also known as microSDHC. We recommend cards with Application Performance Class A1.
- A microSD card slot or card reader in your computer

  If your computer does not have a (micro) SD card slot, you can use an external USB (micro) SD card reader.
- A mini-USB cable, like the one included with your EV3 set

The typical configuration of this equipment is summarized in Figure 1.1.

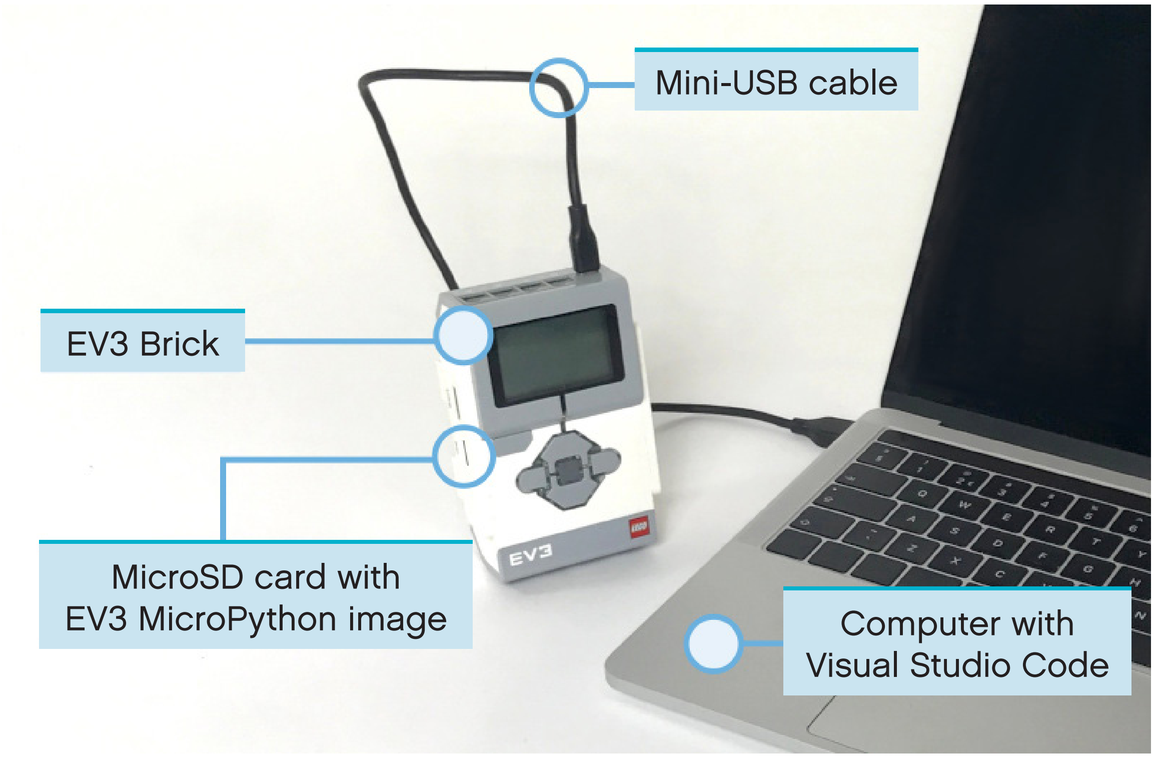

Mini-USB cable
EV3 Brick
MicroSD card with EV3 MicroPython image
Computer with Visual Studio Code

Figure 1.1: Setup overview

1.2 Preparing your computer

You'll write your MicroPython programs using Visual Studio Code. Follow the steps below to download, install, and configure this application:

1. Download Visual Studio Code.
2. Follow the on-screen instructions to install the application.
3. Launch Visual Studio Code.
4. Open the extensions tab.
5. Install the EV3 MicroPython extension as shown in Figure 1.2.

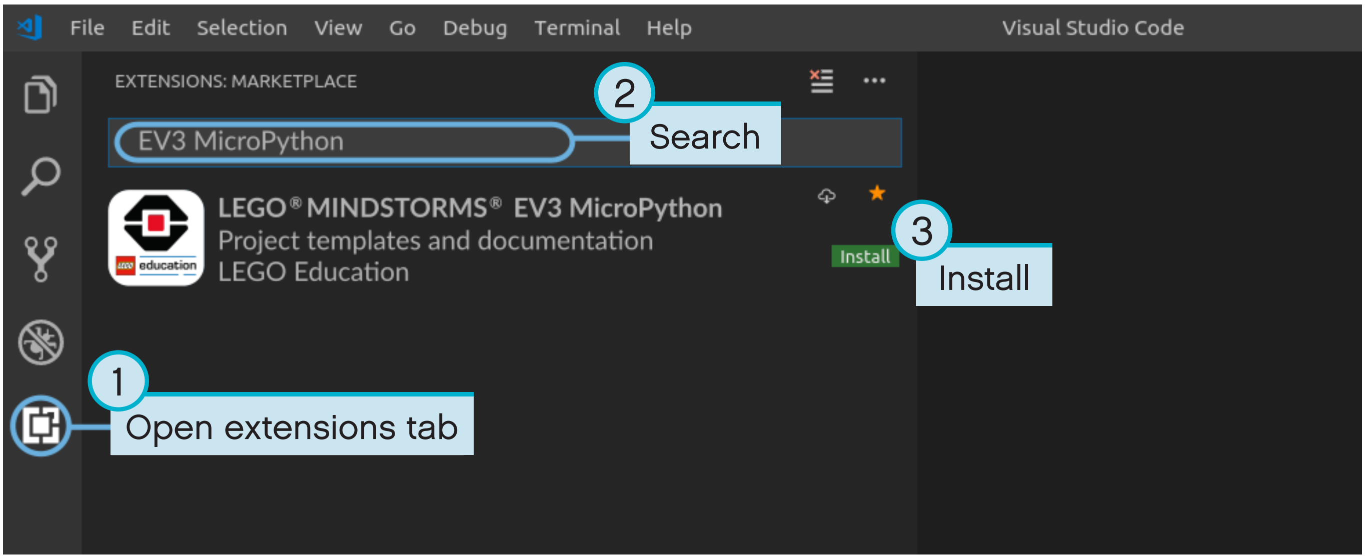

File Edit Selection View Go Debug Terminal Help
Visual Studio Code
EXTENSIONS: MARKETPLACE
EV3 MicroPython
2
Search
LEGO® MINDSTORMS® EV3 MicroPython
Project templates and documentation
LEGO Education
Install
3
Install
1
Open extensions tab

Figure 1.2: Installing the extension from the Visual Studio Code marketplace

1.3 Preparing the microSD card

To make it possible to run MicroPython programs on your EV3 Brick, you'll now learn how to install the required tools on your microSD card.

If the microSD card contains files you want to keep, make sure to create a backup of its contents first. See 'Managing files on the EV3 Brick' to learn how to backup your previous MicroPython programs if necessary.

This process erases everything on your microSD card, including any previous MicroPython programs on it.

To install the MicroPython tools on your microSD card:

1. Download the EV3 MicroPython microSD card image and save it in a convenient location. This file is approximately 360 MB. You do not need to unzip the file.
2. Download and install a microSD card flashing tool such as Etcher.
3. Insert the microSD card into your computer or card reader.
4. Launch the flashing tool and follow the steps on your screen to install the file you have just downloaded. If you use Etcher, you can follow the instructions below, as shown in Figure 1.3.
   a. Select the EV3 MicroPython microSD card image file you have just downloaded.
   b. Select your microSD card. Make sure that the device and size correspond to your microSD card.
   c. Start the flashing process. This may take several minutes. Do not remove the card until the flashing process is complete.

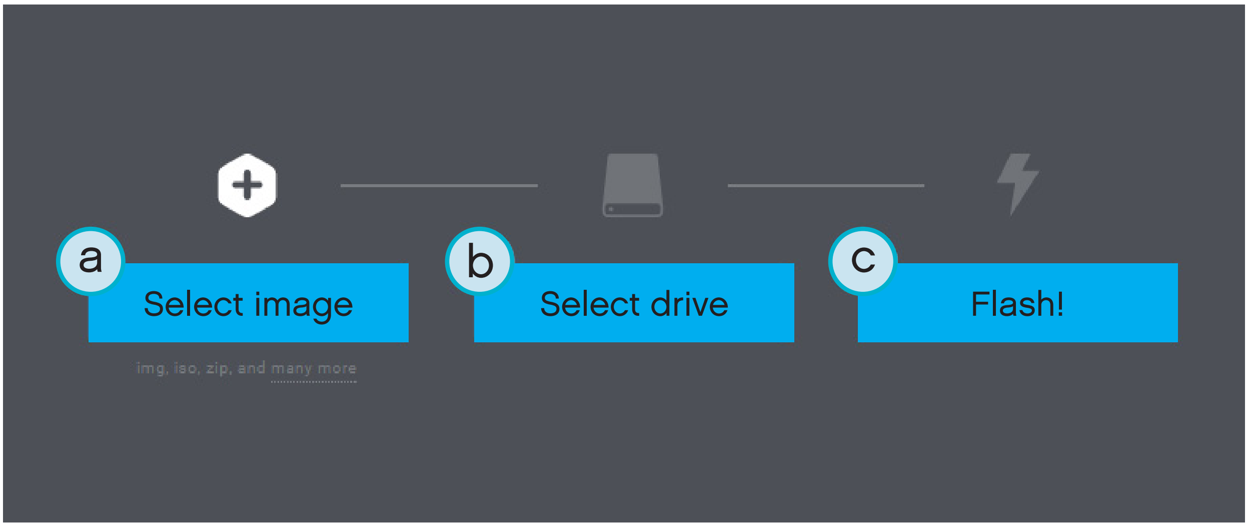


Figure 1.3: Using Etcher to flash the EV3 MicroPython microSD card image

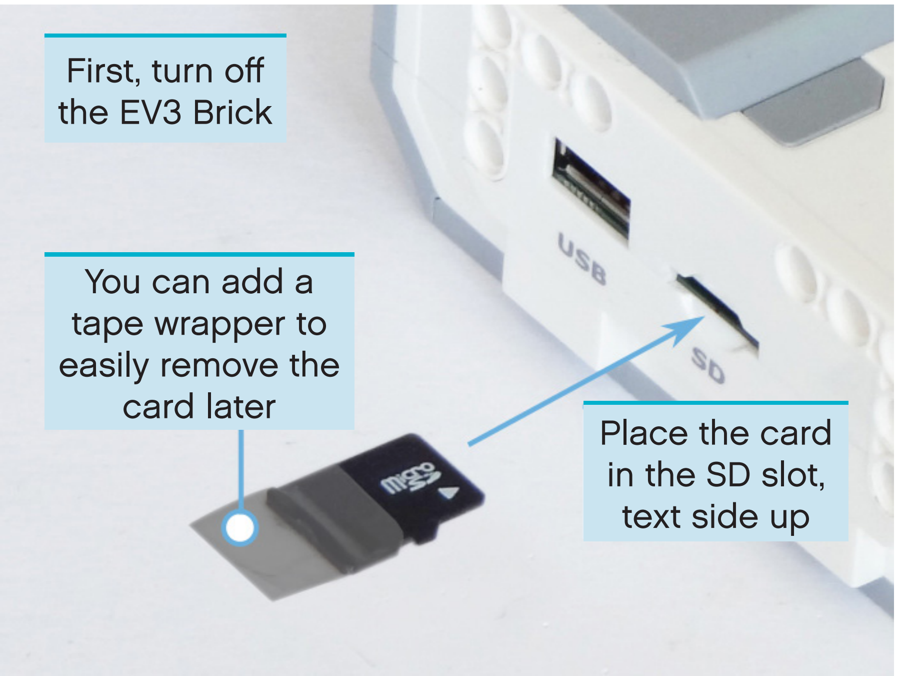
1.4 Updating the microSD card

To update the microSD card, download a new image file using the link above and flash it to the microSD card as described above. Be sure to back up any MicroPython programs you want to save.

You do not need to erase the contents of the microSD card first. This is done automatically when you flash the new image file.

1.5 Using the EV3 Brick

Make sure the EV3 Brick is turned off. Insert the microSD card you prepared into the microSD card slot on the EV3 Brick, as shown in Figure 1.4.


First, turn off the EV3 Brick

You can add a tape wrapper to easily remove the card later

Place the card in the SD slot, text side up

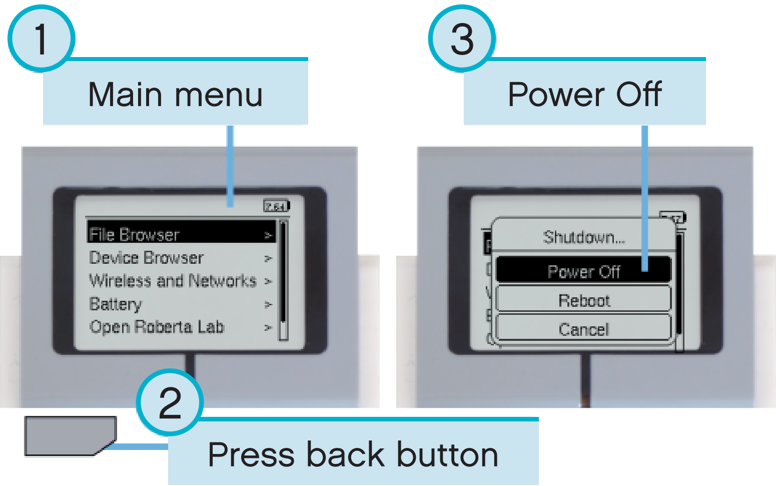
Figure 1.4: Inserting the flashed microSD card into the EV3 Brick

1.5.1 Turning the EV3 Brick on and off

Turn on the EV3 Brick by pressing the dark gray center button.

The boot process may take several minutes. While booting, the EV3 Brick status light turns orange and blinks intermittently, and you'll see a lot of text on the EV3 Display. The EV3 Brick is ready for use when the status light turns green.

To turn the EV3 Brick off, open the shutdown menu with the back button, and then select Power Off using the center button, as shown in Figure 1.5.


1 Main menu

2 Press back button

3 Power Off

Figure 1.5: Turning the EV3 Brick off

1.5.2 Viewing motor and sensor values

When you’re not running a program, you can view motor and sensor values using the device browser, as shown in Figure 1.6.

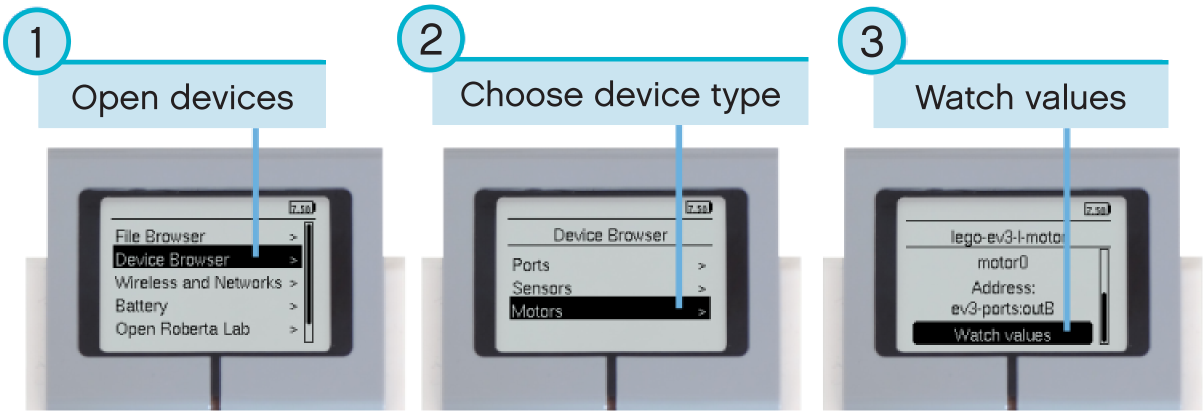


Figure 1.6: Viewing motor and sensor values

1.5.3 Going back to the original firmware

You can go back to the LEGO® firmware and your LEGO programs at any time. To do so:

1. Turn the EV3 Brick off as shown above.
2. Wait for the Display and brick status light to turn off.
3. Remove the microSD card.
4. Turn the EV3 on.

CHAPTER

TWO

CREATING AND RUNNING PROGRAMS

Now that you've set up your computer and EV3 Brick, you're ready to start writing programs.

To make it easier to create and manage your programs, let's first have a quick look at how MicroPython projects and programs for your EV3 robots are organized.

Programs are organized into project folders, as shown in Figure 2.1. A project folder is a directory on your computer that contains the main program (**main.py**) and other optional scripts or files. This project folder and all of its contents will be copied to the EV3 Brick, where the main program will be run.

This page shows you how to create such a project and how to transfer it to the EV3 Brick.

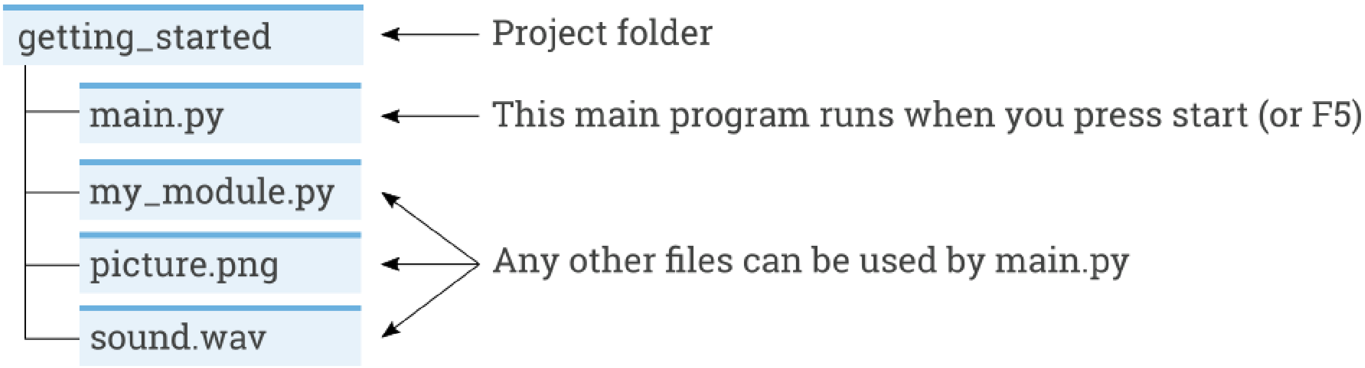
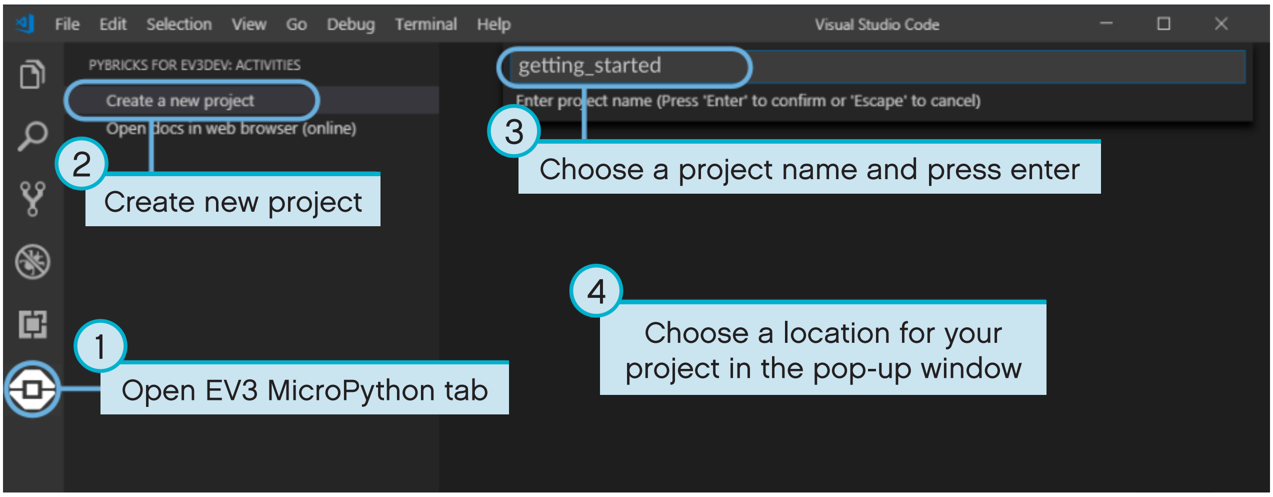

getting_started
Project folder
main.py
This main program runs when you press start (or F5)
my_module.py
picture.png
sound.wav
Any other files can be used by main.py

Figure 2.1: A project contains a program called **main.py** and optional resources like sounds or MicroPython modules.

2.1 Creating a new project

To create a new project, open the EV3 MicroPython tab and click *create a new project*, as shown in Figure 2.2. Enter a project name in the text field that appears and press *Enter*. When prompted, choose a location for this program and confirm by clicking *choose folder*.


File Edit Selection View Go Debug Terminal Help
Visual Studio Code
PYBRICKS FOR EV3DEV: ACTIVITIES
Create a new project
Open docs in web browser (online)

getting_started

Enter project name (Press 'Enter' to confirm or 'Escape' to cancel)

1 Open EV3 MicroPython tab

2 Create new project

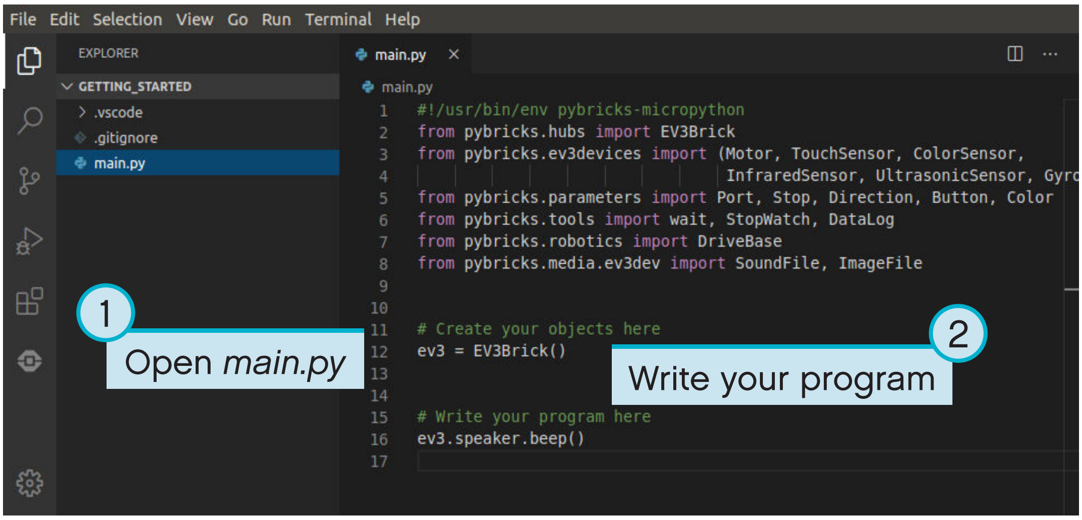
3 Choose a project name and press enter

4 Choose a location for your project in the pop-up window

Figure 2.2: Creating a new project. This example is called *getting_started*, but you can choose any name.

When you create a new project, it already includes a file called *main.py*. To see its contents and to modify it, open it from the file browser as shown in Figure 2.3. This is where you'll write your programs.

If you are new to MicroPython programming, we recommend that you keep the existing code in place and add your code to it.



```
#!/usr/bin/env pybricks-micropython
from pybricks.hubs import EV3Brick
from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
                                 InfraredSensor, UltrasonicSensor, Gyro
from pybricks.parameters import Port, Stop, Direction, Button, Color
from pybricks.tools import wait, StopWatch, DataLog
from pybricks.robotics import DriveBase
from pybricks.media.ev3dev import SoundFile, ImageFile


# Create your objects here
ev3 = EV3Brick()


# Write your program here
ev3.speaker.beep()
```

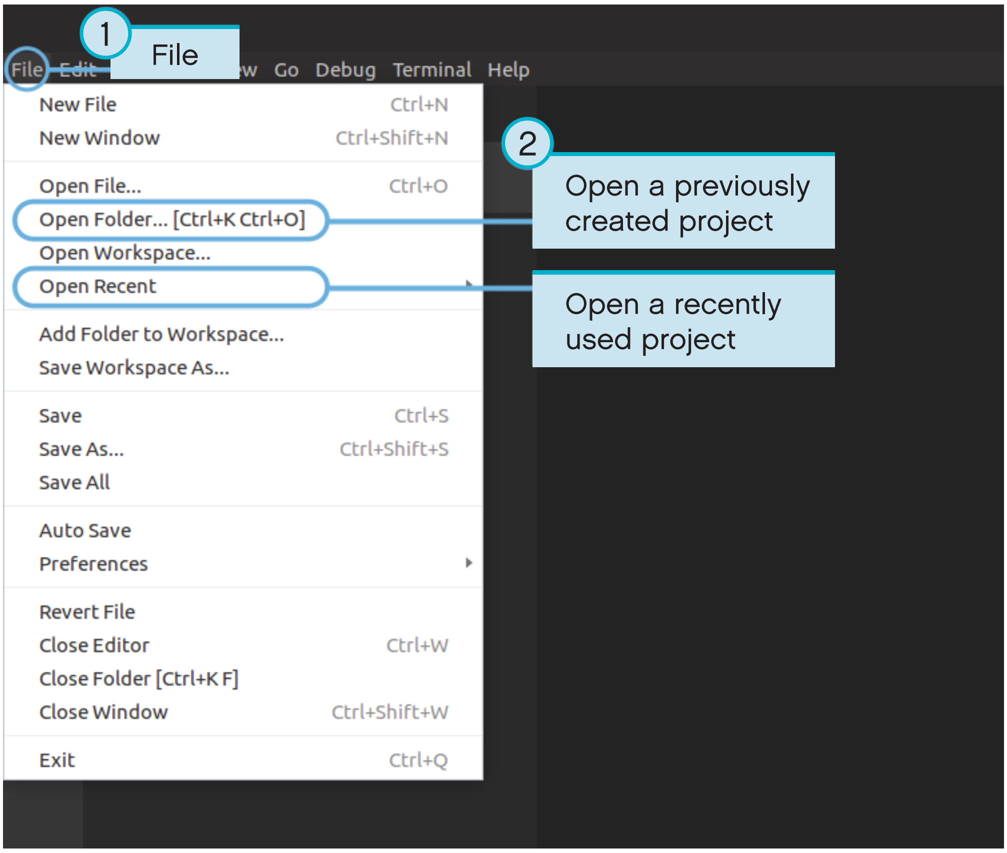
1 Open *main.py*

2 Write your program

Figure 2.3: Opening the default main.py program.

2.2 Opening an existing project

To open a project you created previously, click *File* and click *Open Folder*, as shown in Figure 2.4. Next, navigate to your previously created project folder and click *OK*. You can also open your recently used projects using the *Open Recent* menu option.



1 File

2 Open a previously created project

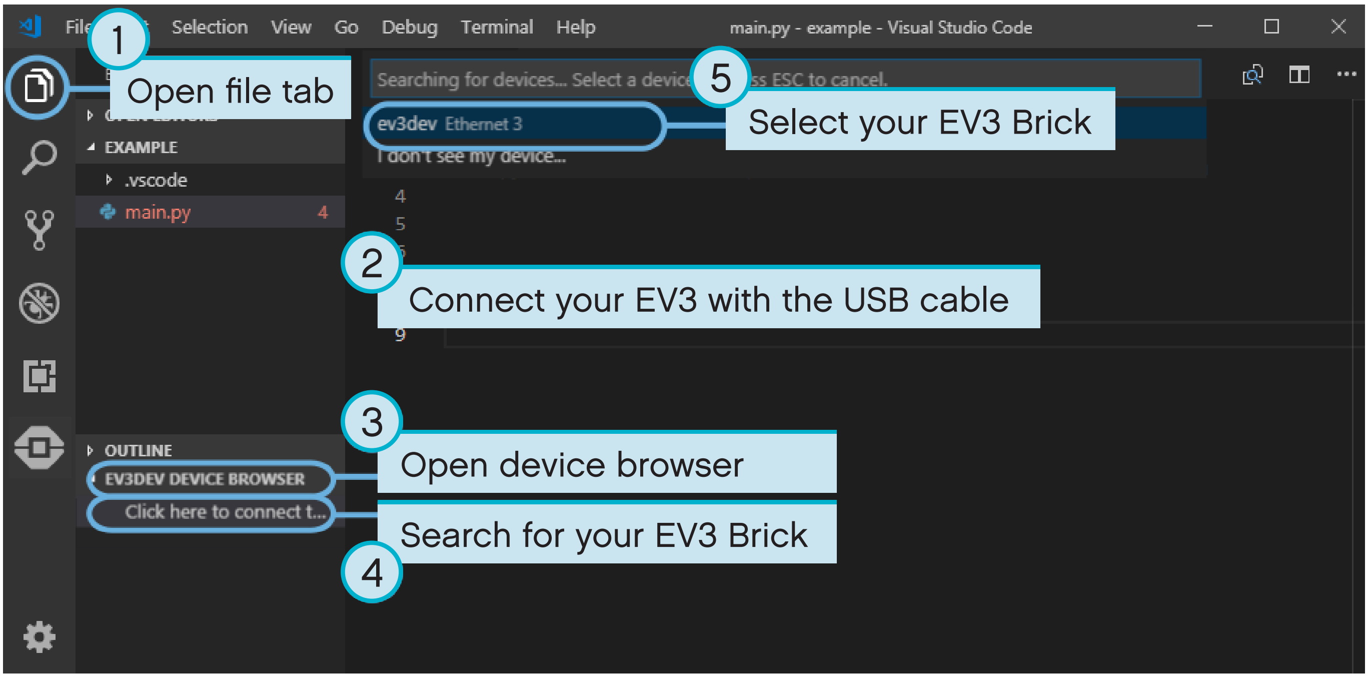
Open a recently used project

Figure 2.4: Opening a previously created project.

2.3 Connecting to the EV3 Brick with Visual Studio Code

To be able to transfer your code to the EV3 Brick, you'll first need to connect the EV3 Brick to your computer with the mini-USB cable and configure the connection with Visual Studio Code. To do so:

- Turn the EV3 Brick on
- Connect the EV3 Brick to your computer with the mini-USB cable
- Configure the USB connection as shown in Figure 2.5.



Searching for devices... Select a device ... ss ESC to cancel.

1 Open file tab

2 Connect your EV3 with the USB cable

3 Open device browser

4 Search for your EV3 Brick

5 Select your EV3 Brick

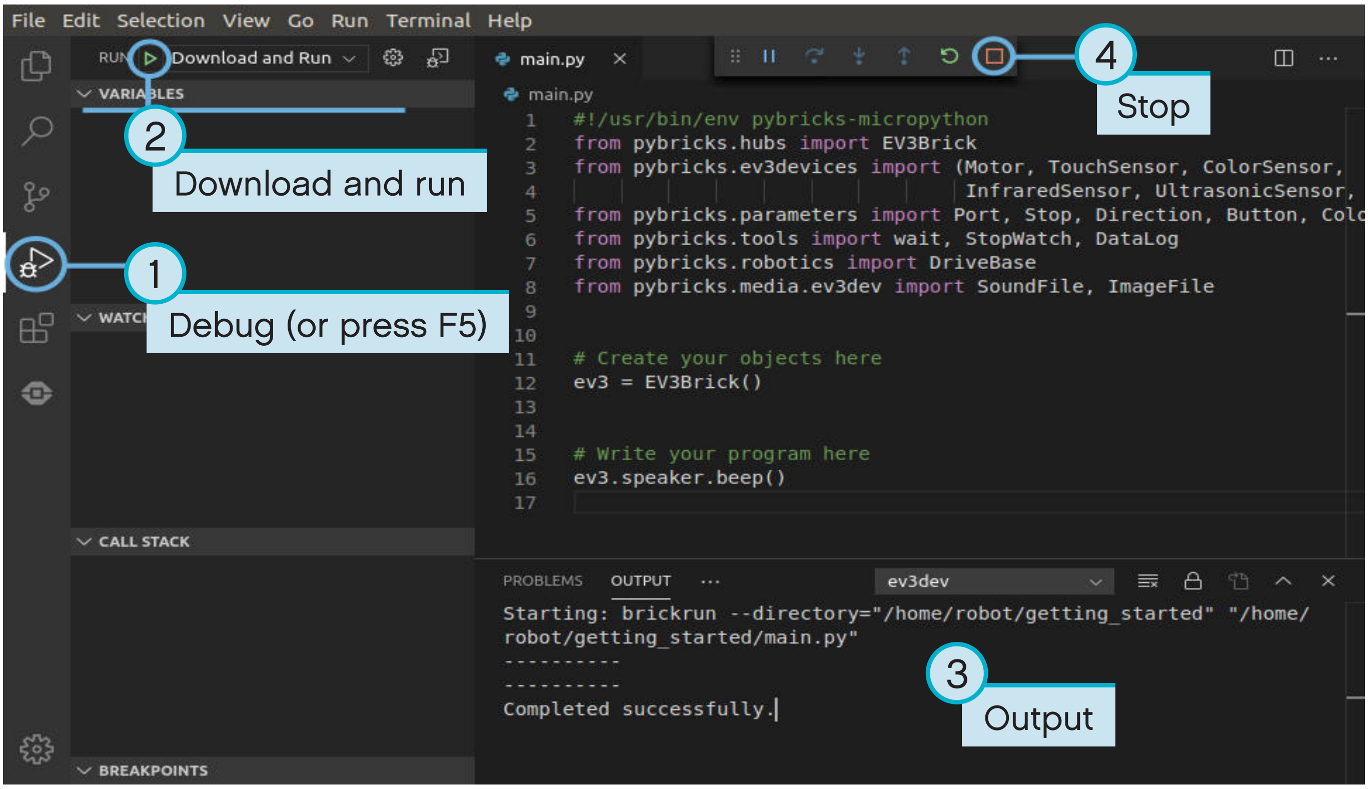
Figure 2.5: Configuring the USB connection between the computer and the EV3 Brick.

2.4 Downloading and running a program

You can press the F5 key to run the program. Alternatively, you can start it manually by going to the debug tab and clicking the green start arrow, as shown in Figure 2.6.

When the program starts, a pop-up toolbar allows you to stop the program if necessary. You can also stop the program at any time using the back button on the EV3 Brick.

If your program produces any output with the *print* command, this is shown in the output window.



```
#!/usr/bin/env pybricks-micropython
from pybricks.hubs import EV3Brick
from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
                                 InfraredSensor, UltrasonicSensor,
from pybricks.parameters import Port, Stop, Direction, Button, Colo
from pybricks.tools import wait, StopWatch, DataLog
from pybricks.robotics import DriveBase
from pybricks.media.ev3dev import SoundFile, ImageFile


# Create your objects here
ev3 = EV3Brick()


# Write your program here
ev3.speaker.beep()
```

```
Starting: brickrun --directory="/home/robot/getting_started" "/home/
robot/getting_started/main.py"
----------
----------
Completed successfully.
```

Figure 2.6: Running a program.

2.5 Expanding the example program

Now that you've run the basic code template, you can expand the program to make a motor move. First, attach a Large Motor to Port B on the EV3 Brick, as shown in Figure 2.7.

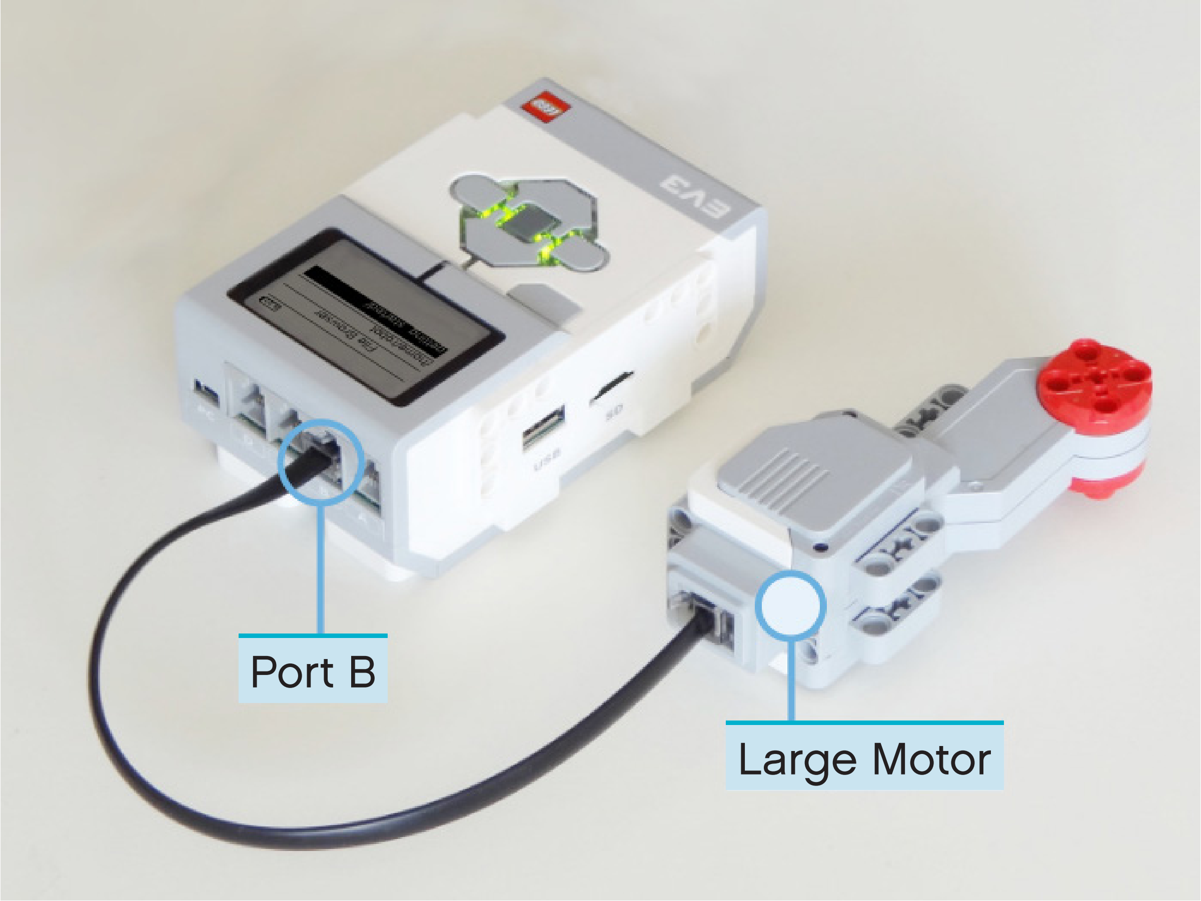


Figure 2.7: The EV3 Brick with a Large Motor attached to port B.

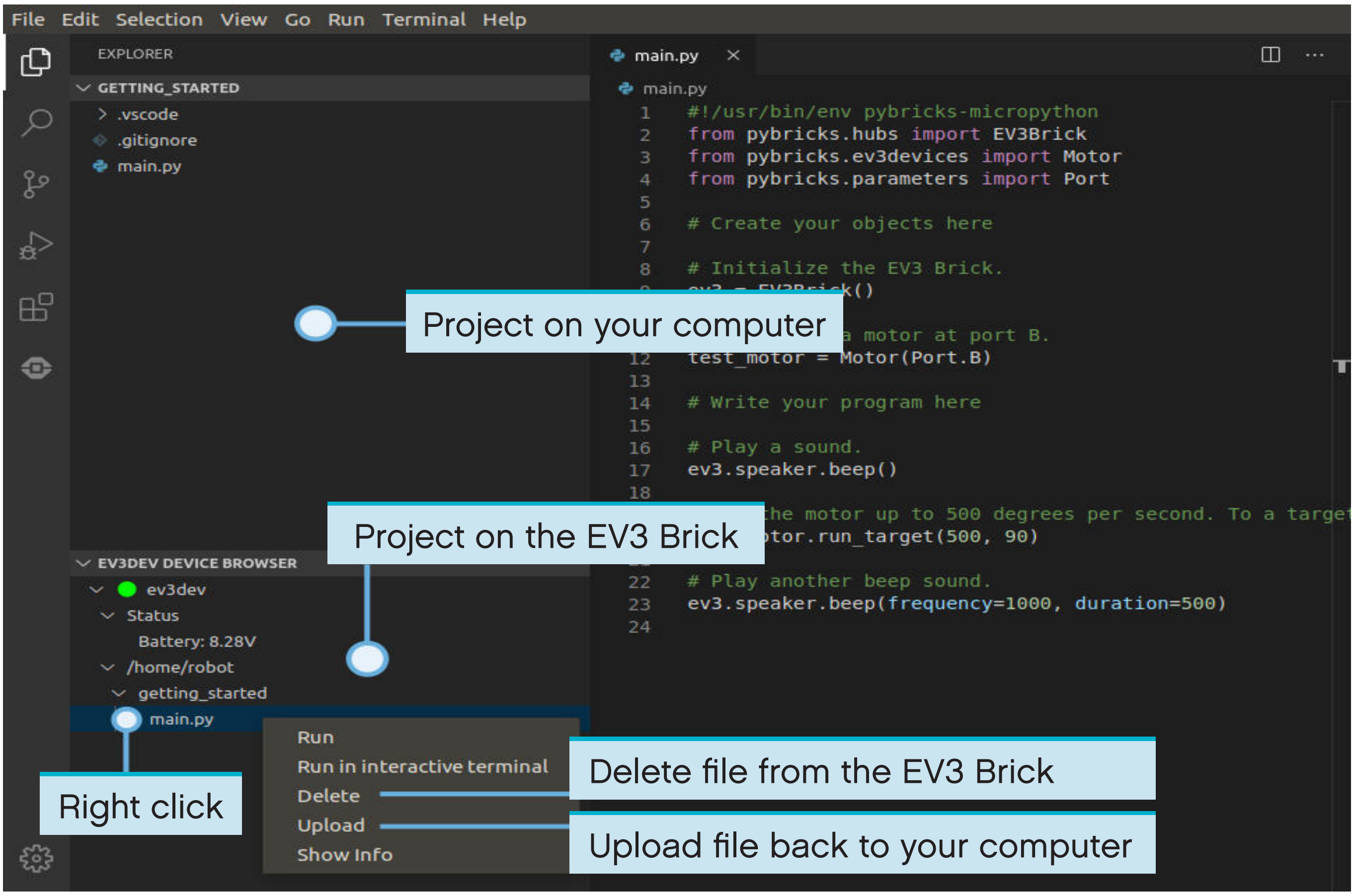
Next, edit *main.py* to make it look like this:

```
#!/usr/bin/env pybricks-micropython

from pybricks.hubs import EV3Brick
from pybricks.ev3devices import Motor
from pybricks.parameters import Port

# Initialize the EV3 brick.
ev3 = EV3Brick()
# Initialize a motor at port B.
test_motor = Motor(Port.B)

# Play a sound.
ev3.speaker.beep()
# Run the motor up to 500 degrees per second. To a target angle of 90 degrees.
test_motor.run_target(500, 90)
# Play another beep sound.
ev3.speaker.beep(1000, 500)
```

This program makes your robot beep, rotate the motor, and beep again with a higher pitched tone. Run the program to make sure that it works as expected.

2.6 Managing files on the EV3 Brick

After you've downloaded a project to the EV3 Brick, you can run, delete, or back up programs stored on it using the device browser as shown in in Figure 2.8.



main.py

```
#!/usr/bin/env pybricks-micropython
from pybricks.hubs import EV3Brick
from pybricks.ev3devices import Motor
from pybricks.parameters import Port

# Create your objects here

# Initialize the EV3 Brick.
ev3 = EV3Brick()

                  a motor at port B.
test_motor = Motor(Port.B)

# Write your program here

# Play a sound.
ev3.speaker.beep()

             the motor up to 500 degrees per second. To a target
             otor.run_target(500, 90)

# Play another beep sound.
ev3.speaker.beep(frequency=1000, duration=500)

```

Figure 2.8: Using the EV3 device browser to manage files on your EV3 Brick.

2.7 Running a program without a computer

You can run previously downloaded programs directly from the EV3 Brick.

To do so, find the program using the *file browser* on the EV3 Display and press the center button key to start the program as shown in in Figure 2.9.

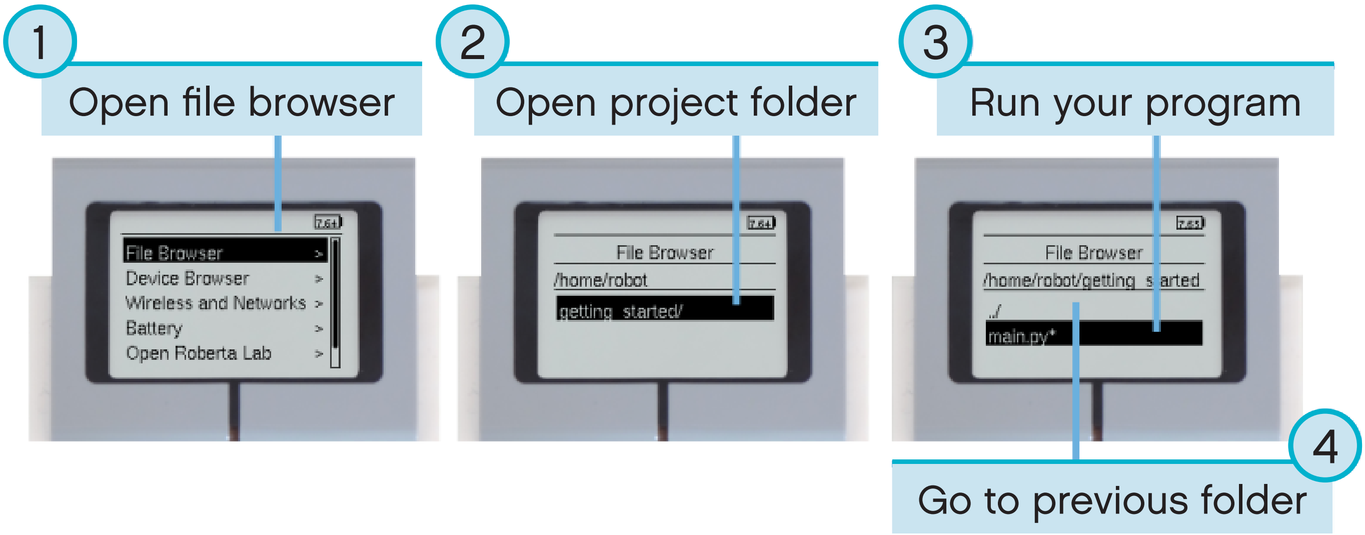


Figure 2.9: Starting a program using the buttons on the EV3 Brick.