



Перейдите на новый уровень в обучении предметам STEAM

Используйте наборы LEGO® MINDSTORMS® Education EV3 совместно с любой средой программирования, чтобы усовершенствовать и дополнить свои программы обучения по предметам естественнонаучного и технического циклов (STEAM), а также робототехнике в старших классах основной школы и учреждениях среднего профессионального образования.

Теперь платформа EV3 поддерживает среду программирования MicroPython. Опыт работы с этим ПО позволит ученикам приобрести необходимые навыки программирования для построения успешной карьеры в будущем.

Подготовка учеников к дальнейшему обучению и профессиональному росту

Благодаря ПО MicroPython учащиеся и педагоги смогут использовать расширенные возможности конструкторов EV3 для решения задач из реальной жизни. Планы уроков с использованием наборов EV3, в основу которых легли технологии автоматизированного производства и проектирования реально существующих транспортных средств, были специально разработаны, чтобы помочь ученикам старших классов развить навыки, необходимые для успешного построения карьеры. Предлагаемые материалы сфокусированы на изучении современных информационных технологий, методов инженерного проектирования и технологий автоматизации производства.



Начало работы

Чтобы испытать все возможности программирования роботов EV3 на языке Python, просто установите символ EV3 MicroPython на микро-SD-карту и используйте её для загрузки микрокомпьютера EV3. Установите редактор Visual Studio Code от Microsoft на свой компьютер и найдите расширение EV3 MicroPython. Можно начинать программировать!

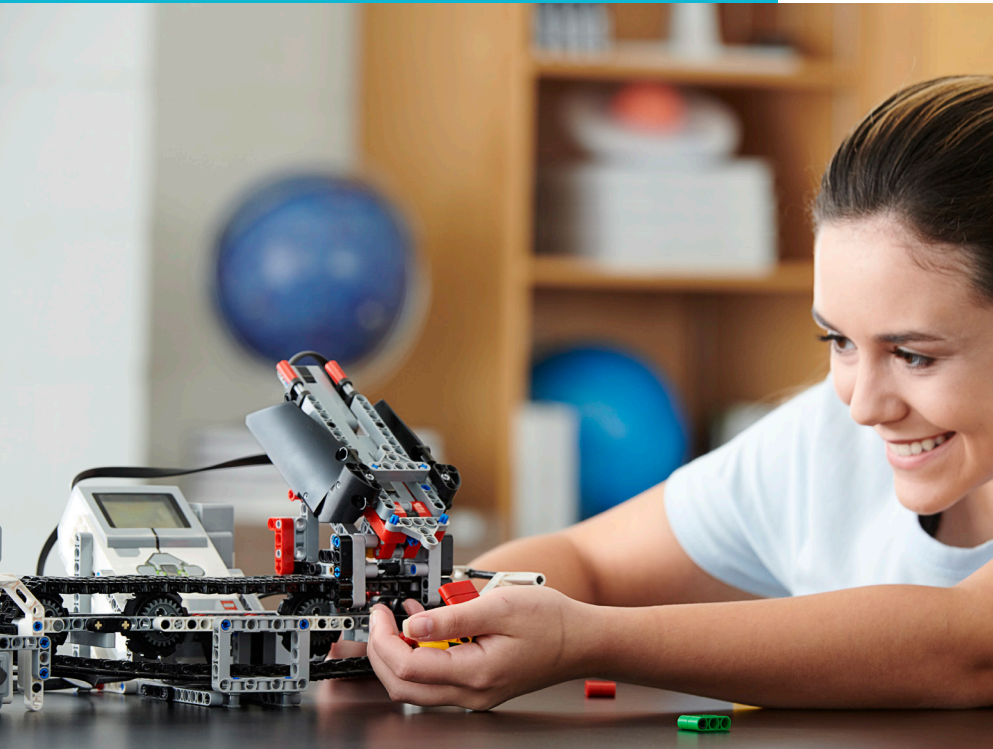
Пользователям доступно руководство по началу работы, содержащее подробную информацию об установке ПО, документацию по программированию, а также простые программы и справочные материалы по общим вопросам.



Микрокомпьютер EV3

Карта Micro SD с прошивкой EV3 Python

Компьютер с установленным редактором Visual Studio Code



Что такое MicroPython?

MicroPython — это версия языка программирования Python, оптимизированная для работы на микроконтроллерах, подобных микрокомпьютеру EV3. MicroPython помогает ученикам старших классов приобрести навыки, которые необходимы для эффективного программирования на языке Python на профессиональном уровне.

Ученики смогут писать программы на языке MicroPython в редакторе с открытым исходным кодом Visual Studio Code, разработанным Microsoft для Windows, MacOS и Linux.

Для получения более подробной информации и поддержки посетите веб-сайт LEGO® MINDSTORMS® Education EV3:

<https://education.lego.com/ru-ru/support/mindstorms-ev3/python-for-ev3>

```
1 #!/usr/bin/env python3
2
3 from pybricks import ev3brick as brick
4 from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
5                                 GyroSensor)
6 from pybricks.parameters import Port, Stop, Direction, Color, ImageFile
7 from pybricks.tools import wait
8
9 # configure the turntable motor, which rotates the arm. It has a
10 # 20 tooth, a 12 tooth and a 28 tooth gear connected to it.
11 turntable_motor = Motor(Port.B, Direction.CLOCKWISE, [20, 12, 28])
12
13 # Configure the seesaw motor with default settings. This motor raises
14 # and lowers the Pen Holder.
15 seesaw_motor = Motor(Port.C)
16
17 # Set up the Gyro Sensor. It is used to measure the angle of the arm.
18 # Keep the Gyro Sensor and EV3 steady when connecting the cable and
19 # during start-up of the EV3.
20 gyro_sensor = GyroSensor(Port.S2)
21
22 # Set up the Color Sensor. It is used to detect whether there is white
23 # paper under the drawing machine.
24 color_sensor = ColorSensor(Port.S3)
25
26 # Set up the Touch Sensor. It is used to detect when it is pressed,
27 # holding it to start drawing the pattern.
28 touch_sensor = TouchSensor(Port.S4)
29
30 def pen_holder_raise():
31     # This function raises the Pen Holder.
32     seesaw_motor.run_target(50, 25, Stop.HOLD)
33     wait(1000)
34
35 def pen_holder_lower():
36     # This function lowers the Pen Holder.
37     seesaw_motor.run_target(50, 0, Stop.HOLD)
38     wait(1000)
39
40 def pen_holder_turn_to(target_angle):
41     # This function turns the arm to the specified target angle.
42
43     # Run the turntable motor until the arm reaches the target angle.
44     if target_angle > gyro_sensor.angle():
45         # If the target angle is greater than the current Gyro Sensor
46         # angle, run clockwise at a positive speed.
47         turntable_motor.run(70)
48         while gyro_sensor.angle() < target_angle:
49             pass
50     elif target_angle < gyro_sensor.angle():
51         # If the target angle is less than the current Gyro Sensor
52         # angle, run counter-clockwise at a negative speed.
53         turntable_motor.run(-70)
54         while gyro_sensor.angle() > target_angle:
55             pass
56     # Stop the motor when the target angle is reached.
57     turntable_motor.stop(0, 0, 0)
```

