

Дарья Михайловна Гребнева
г. Нижний Тагил

Программирование учебных роботов на базе Lego Education EV3 на языке программирования Python

В статье описаны основы управления учебными роботами на языке программирования Python. Актуальность использования Python в образовательной робототехнике заключается в том, что он делает работу с робототехническими наборами ближе к задачам из реальной жизни, что способствует мотивации обучающихся к изучению робототехники и программирования. В статье приведен обзор основных команд языка Python для управления учебными роботами на базе платформы Lego Education EV3, а также описаны практические примеры реализации типовых алгоритмов управления учебными роботами. Рассмотрены три примера: вывод данных на дисплей, движение мобильного робота с одним датчиком цвета по черной линии, движение мобильного робота с двумя датчиками цвета по черной линии. Статья может быть интересна педагогам дополнительного образования, а также обучающимся, которые интересуются образовательной робототехникой на базе платформы Lego Education EV3.

Ключевые слова: образовательная робототехника, Lego Education EV3, язык программирования Python, расширение EV3 MicroPython, управление учебными роботами.

Daria Mikhailovna Grebneva
Nizhny Tagil

Programming educational robots based on Lego Education EV3 in the Python programming language

This article describes the controlling educational robots' basics in the Python programming language. The relevance of using Python in educational robotics is that it makes working with robotic sets closer to real-life tasks which helps to motivate students to learn robotics and programming. The article provides an overview of the main Python commands for controlling educational robots based on the Lego Education EV3 platform and also describes practical examples of implementing typical algorithms for controlling educational robots. Three examples were considered: data output to the display, the movement of a mobile robot with one color sensor along the black line and the movement of a mobile robot with two color sensors along the black line. The article may be of interest to teachers of additional education as well as students who are interested in educational robotics on the Lego Education EV3 platform.

Keywords: educational robotics, Lego Education EV3, Python programming language, EV3 MicroPython extension, teaching robot control.

Базовая среда программирования EV3 создана на основе графического языка LabVIEW и позволяет задавать алгоритмы для робота визуально: команды представлены в виде блоков, которые можно перетаскивать и соединять. Такой способ хорошо работает, когда нужно показать, как строятся алгоритмы, но он не подходит для программ с большим количеством блоков. При усложнении сценариев необходимо переходить на программирование с помощью кода [1].

Среда EV3 построена на базе процессора ARM9 с открытой архитектурой. Это решение позволяет устанавливать альтернативные прошивки, в частности образ для работы с MicroPython [2]. Это даёт возможность использовать Python для программирования микрокомпьютера EV3, что приближает работу с набором к решению задач из реальной жизни.

Поддержка MicroPython делает платформу EV3 подходящей для обучения в старших классах. Обучающиеся могут попробовать себя в роли программистов на одном из самых популярных современных языков, познакомиться с профессиями, связанными с программированием и инженерным проектированием.

Для начала программирования в среде MicroPython необходимо установить редактор кода Visual Studio Code с расширением EV3 MicroPython, а также установить образ с EV3 MicroPython на SD-карту.

После подготовки необходимых инструментов можно приступать непосредственно к программированию. Основные команды управления роботом на базе Lego Ev3 на языке Python представлены в таблице 1.

Таблица 1

Основные команды управления роботом на базе Lego EV3 на языке Python

Команда	Действие	Пример
buttons ()	Управление кнопками	<pre># Вывод надписи на дисплей при нажатии левой кнопки на блоке if Button.LEFT in brick.buttons(): print("The left button is pressed") # Ожидание нажатия любой из кнопки while any(brick.buttons()): wait(10)</pre>

Команда	Действие	Пример
light(color)	Подсветка кнопок микрокомпьютера	# Подсветка красным цветом brick.light(Color.RED) # Выключение подсветки brick.light(None)
sound.beep(frequency, duration, volume)	Звуковой сигнал	# Короткий звуковой сигнал brick.sound.beep() # Звуковой сигнал с заданными настройками (частота звучания 1500Гц, время звучания 1 с. (1000 мс.), громкость 50%) brick.sound.beep(1500, 1000, 50)
clear()	Очистка дисплея	display.clear()
text(text, coordinate)	Отображение заданного текста на дисплее	# Вывод надписи "Start", начиная с точки с координатами (50,60) display.text("Start", (50,60))
display.image(file_name, alignment, coordinate, clear)	Отображение векторного рисунка на дисплее	# Отображение векторного рисунка с заданными параметрами display.image("example.png", alignment=Align.CENTER, coordinate=None, clear=True)
battery.voltage()	Получение данных о напряжении батареи.	# Если напряжение менее 7 В (7000 мВ), то подать звуковой сигнал if battery.voltage() < 7000: brick.sound.beep()
Motor(port, direction, gears)	Управление мотором	# Инициализация мотора, подключенного к порту В, направление против часовой стрелки right_motor = motor(Port.B, Direction.COUNTERCLOCKWISE)
speed()	Получение данных о скорости вращения мотора	# Вывод значения текущей скорости на дисплей motorC = motor(Port.C) display.text(motorC.speed())
pressed()	Проверка нажатия на датчик.	Returns True при наличии нажатия. False, если нажатие отсутствует. Return type bool
ColorSensor (port)	Инициализация датчика цвета	# Двигаться вперед на скорости 30, пока датчиком не будет определен черный цвет (например, наезд на черную линию). while colorSeen != Color.BLACK: leftMotor.dc(30) rightMotor.dc(30)
InfraredSensor (port)	Инициализация инфракрасного датчика	# Инициализация инфракрасного датчика, подключенного к порту №2 sensorInf = InfraredSensor(Port.2)
UltrasonicSensor (port)	Инициализация ультразвукового датчика	# Инициализация ультразвукового датчика, подключенного к порту №2 sensorInf = InfraredSensor(Port.2)

Для иллюстрации действия команд рассмотрим несколько практических примеров написания программ для управления движением учебными роботами.

Пример 1. Рассмотрим пример написания простейшей программы на языке Python, которая выводит на дисплей значение текущей зарядки батареи блока Lego Ev3 Education (листинг 1).

Листинг 1

```
# Инициализировать библиотеку модулей EV3 и подключить необходимые компоненты:
#!/usr/bin/env pybricks-micropython
from pybricks import ev3brick as brick
from pybricks.tools import wait
#Подать короткий звуковой сигнал при запуске программы
brick.sound.beep()
#Очистить дисплей от предыдущих данных
brick.display.clear()
#Показать на дисплее текущий заряд батареи
brick.display.text("Voltage is: {}".format(brick.battery.voltage()))
#Ждать пока не нажата какая-либо кнопка на блоке
```

```
while not brick.buttons():  
    wait(10)
```

В структуру программы входит инициализация библиотеки Ev3 с подключением необходимых компонентов. Для рассмотренного выше примера достаточно компонента brick для доступа к дисплею и батарее устройства и метода wait(). Далее идет основное тело программы, где предварительно очищается дисплей, выводятся на дисплей данные о текущем заряде батареи.

Пример 2. Рассмотрим реализацию простейшего алгоритма движения мобильного

робота с одним датчиком цвета по черной линии. Алгоритм очень простой: если датчик видит черный цвет, то робот поворачивает в одну сторону, если белый – в другую (листинг 2).

Отметим, что в результате выполнения программы робот будет двигаться не строго по черной линии, а по ее границе, подворачивая то влево, то вправо и постепенно перемещаясь вперед (рис. 1).

Листинг 2

```
while a==1:  
    colorSeen = colorScanned.color()  
    if (colorSeen== Color.BLACK):  
        leftMotor.dc(30)  
        rightMotor.dc(30)  
    else:  
        rightMotor.dc(-30)
```

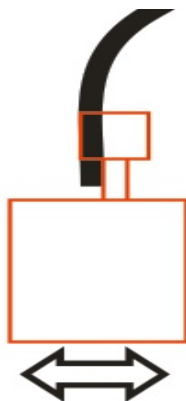


Рис. 1. Движение по черной линии

Пример 3. Рассмотрим движение по линии с использованием двух датчиков цвета.

Для реализации алгоритма нам потребуется отслеживать показания обоих датчиков и только после этого задавать движение мобильному

роботу. Для этого будем использовать вложенные условия: мы опросим сначала первый датчик, а потом, независимо от показаний первого, опросим второй датчик, после чего зададим действие (листинг 3).

Листинг 3

```
#!/usr/bin/env pybricks-micropython  
#from pybricks.ev3devices import ColorSensor  
#colourScanner = ColorSensor(Port.S1)  
#!/usr/bin/env pybricks-micropython  
from pybricks import ev3brick as brick  
from pybricks.parameters import Color, Port  
from pybricks.tools import wait  
from pybricks.ev3devices import ColorSensor, Motor  
  
colorScanned1 = ColorSensor(Port.S1)  
colorScanned2 = ColorSensor(Port.S3)  
colorSeen1 = colorScanned1.color()  
colorSeen2 = colorScanned2.color()  
leftMotor = Motor(Port.B)  
rightMotor = Motor(Port.C)  
a=1  
while a==1:  
    colorSeen1 = colorScanned1.color()  
    colorSeen2 = colorScanned2.color()  
    if (colorSeen1== Color.WHITE and colorSeen2== Color.WHITE):  
        leftMotor.dc(30)  
        rightMotor.dc(30)  
    if (colorSeen2== Color.WHITE and colorSeen1== Color.BLACK):
```

```
leftMotor.dc(30)
rightMotor.dc(-10)
if (colorSeen2== Color.BLACK and colorSeen1== Color.WHITE):
    leftMotor.dc(-10)
    rightMotor.dc(30)
if (colorSeen1== Color.BLACK and colorSeen2== Color.BLACK):
    leftMotor.dc(0)
    rightMotor.dc(0)
brick.sound.beep()
```

Рассмотренные выше примеры показывают, что синтаксис программ на языке Python для управления учебными роботами на базе Lego Education EV3 достаточно прост. Кроме того, программы, написанные на текстовом языке Python, работают гораздо быстрее, чем программы, написанные с помощью визуальных блоков.

Таким образом, робототехнический набор Lego EV3 с прошивкой MicroPython может помочь обучающимся в полной мере освоить навыки программирования на языке Python. В свою очередь, возможности языка программирования Python открывают дополнительные возможности по реализации сложных алгоритмов управления учебными роботами.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Овсяницкая, Л.Ю. Курс программирования робота Lego Mindstorms EV3 в среде EV3 / Л.Ю. Овсяницкая, Д.Н. Овсяницкий, А.Д. Овсяницкий. – Изд. 2-е, перераб. и доп. – Москва : Перо, 2016. – 296 с.
2. Программирование на языке Python с EV3 : офиц. сайт. – 2021. – URL: <https://education.lego.com/ru-ru/product-resources/mindstorms-ev3/материалы-для-педагогов/python-для-ev3> (дата обращения: 28.07.2021). – Текст : электронный.

REFERENCES

1. Ovsjanickaja L.Ju., Ovsjanickij D.N., Ovsjanickij A.D. Kurs programirovanija robota Lego Mindstorms EV3 v srede EV3 [Lego Mindstorms EV3 Robot Programming course in EV3]. Moscow: Pero, 2016. 296 p.
2. Programirovanie na jazyke Python s EV3: ofic. sajt [Python Programming with EV3]. 2021. URL: <https://education.lego.com/ru-ru/product-resources/mindstorms-ev3/materialy-dlja-pedagogov/python-dlja-ev3> (Accessed 28.07.2021).

СВЕДЕНИЯ ОБ АВТОРЕ:

Д.М. Гребнева, кандидат педагогических наук, доцент кафедры информационных технологий, Нижнетагильский государственный социально-педагогический институт (филиал) ФГАОУ ВО «Российский государственный профессионально-педагогический университет», г. Нижний Тагил, Россия, e-mail: grebdash@gmail.com, ORCID: 000-0002-7511-9327.

INFORMATION ABOUT THE AUTHOR:

D.M. Grebneva, Ph.D. in Pedagogy, Associate Professor, Department of Information Technologies, Nizhny Tagil State Social and Pedagogical Institute (branch) of “Russian State Professional Pedagogical University”, Nizhny Tagil, Russia, e-mail: grebdash@gmail.com, ORCID: 000-0002-7511-9327.