

# Программирование контроллера Lego Mindstorms EV3 в различных средах программирования.

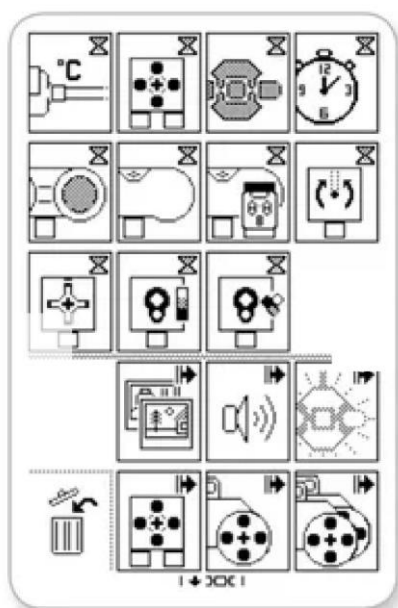
## Введение

Среди робототехнических конструкторов, предназначенных для обучения основам робототехники и программирования, выделяется конструктор, созданный легендарной компанией Lego - Lego Mindstorms EV3. Это, наверное, самый массовый робототехнический конструктор, предназначенный для обучения детей и подростков, начиная от уровня «ничего не умею», и заканчивая уровнем «эксперт». Полная унификация деталей с другими конструкторами Lego, два типа двигателей и несколько датчиков в комплекте, достаточно мощный управляющий микрокомпьютер, работающий на ОС Linux, позволяют создавать очень сложные и интересные проекты. А умение микрокомпьютеров данного конструктора «общаться» между собой в случае необходимости и наличие беспроводных интерфейсов, делают возможности конструктора почти безграничными. Большинство пользователей EV3 уже имеют опыт конструирования на базе классических наборов Lego, поэтому подход фирмы к механической части данного конструктора им хорошо знаком. Чего не скажешь о среде программирования, без которой создание полноценного робота на базе контроллера или микрокомпьютера невозможно. В части выбора софта для создания программ к моделям на базе EV3 вопросы возникают не только у

учащихся, желающих освоить данный конструктор, но и у педагогов, ведущих профильные объединения. Данная статья является кратким обзором программ, позволяющих «оживлять» роботов, созданных из элементов конструктора Lego Mindstorms. Не претендую на обзор всех сред программирования, поддерживающих EV3, выбор обусловлен исключительно моим опытом и предпочтениями.

## 1. Программирование при помощи контроллера.

Самый простой способ, способный заставить робота выполнять простые действия, а также дающий возможность составлять базовые программы, является софт, уже «зашитый» в микрокомпьютере EV3. Для такого программирования не нужны никакие внешние устройства и сторонние среды. Создание программного алгоритма происходит непосредственно на микрокомпьютере, входящим в состав конструктора. После включения управляющего блока и загрузки прошитой в него ОС, мы видим 4 вкладки. Заходим в третью вкладку слева, выбираем пункт Brick Program, и программируем.

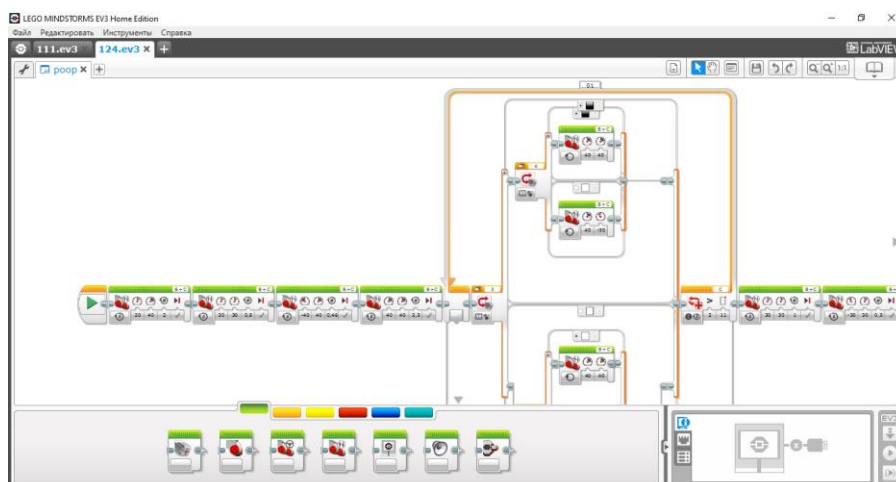


Выбор пиктограмм достаточно ограничен. Экран условно разбит на два блока по функциональной нагрузке, которые можно назвать «Ожидание» (чего либо, это может быть ожидание определенных показаний датчика, ожидание истечения отрезка времени, угла поворота двигателя, и т.д.), и «Действие» (позволяет управлять двигателями, звуком, картинкой экрана и т.д.). Оба блока содержат в сумме 17 модулей, что, в принципе, достаточно для освоения базовых возможностей конструктора.

## 2. Среда программирования Lego Mindstorms EV3.

Следующей средой, являющейся по факту стандартной для программирования роботов, созданных на базе рассматриваемого набора, будет рассмотрена среда Lego Mindstorms EV3. Это яркий пример блочно-визуального программирования при помощи пиктограмм, наиболее наглядного способа, понятного каждому ребенку, умеющему читать и выполнять базовые математические действия.

Нужно отметить, что данная программа основана на очень серьезной среде LabView, созданной в далеком 1986 году компанией National Instruments (США), и использующейся в промышленной робототехнике, АСУТП и еще целом ряде «взрослых» устройств и систем.



Как мы видим, пиктограммы сгруппированы в 6 вкладок внизу рабочей области экрана. Каждая вкладка выделена своим цветом, а группировка осуществляется по функциональному принципу. В данной среде возможна гораздо более гибкая, чем в Brick Program, настройка программы, не говоря уж о гораздо большем количестве пиктограмм в арсенале. Однако, не смотря на обилие инструментов, данный способ программирования EV3 считается наиболее понятным для учащихся.

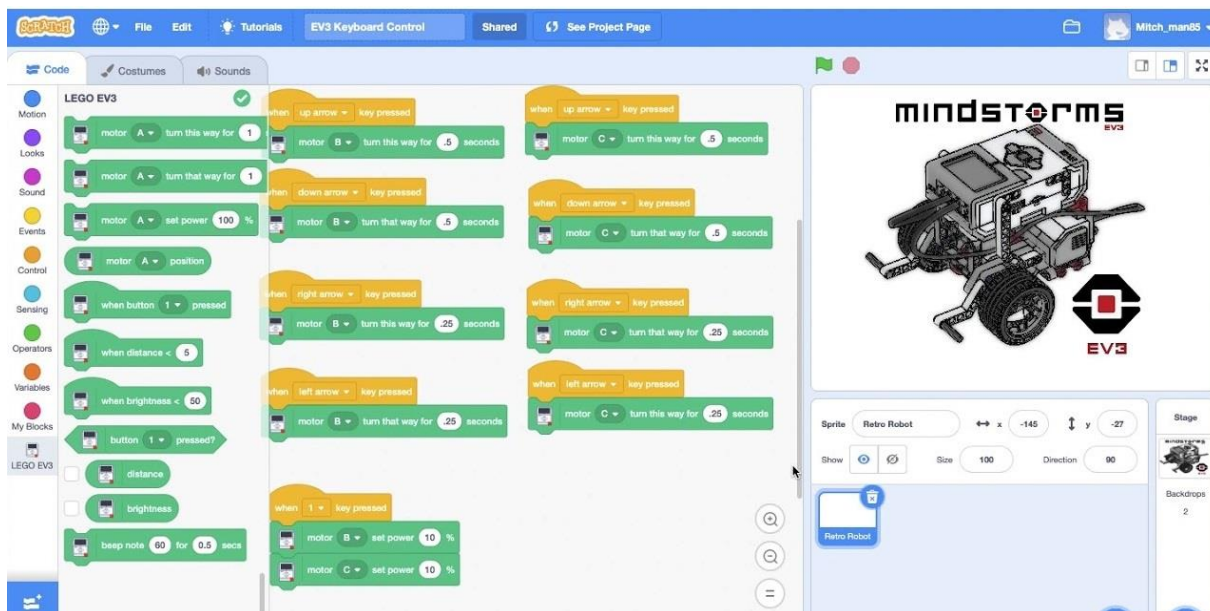
### **3. Scratch и Scratch-подобные среды.**

Более универсальными средами, позволяющими создавать программы для Mindstorms EV3, и которые мне хотелось бы рассмотреть – это Scratch и Scratch-подобные среды.

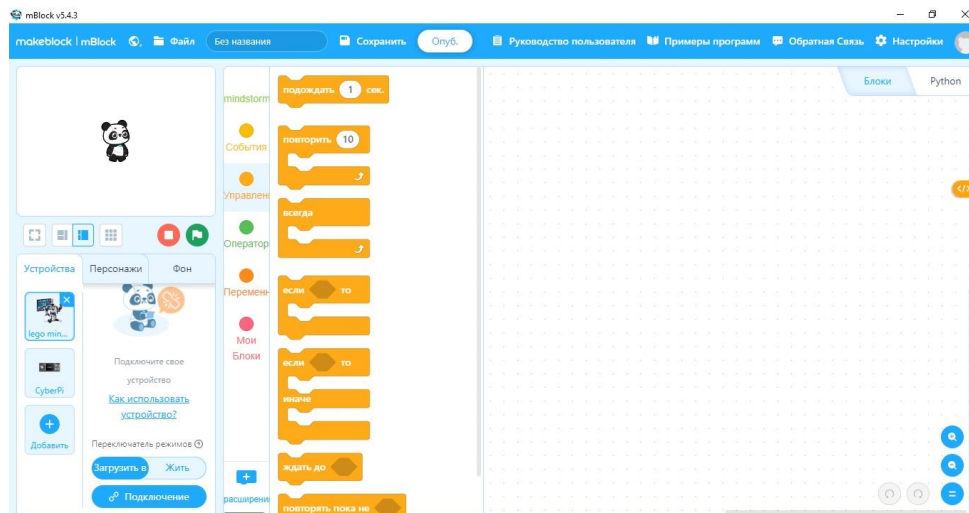
Scratch задумывался как программная платформа, позволяющая детям даже младшего школьного возраста получить опыт создания анимации, игр, или других программ без глубокого погружения в классическое программирование. Вместе с тем, дети, работающие в Scratch, осваивают принципы построения алгоритмов, и, с повышением уровня образования, уже подготовленными могут переходить к консольному программированию или системам визуального программирования более высокого уровня. Пилотная версия приложения вышла в 2003 году, и практически сразу Scratch стал очень популярным. С развитием проекта появилось много надстроек, включая те, что позволяют работать с различными контроллерами. В данной среде программирования реализована так близкая компании Lego идеология разработки посредством комбинации стандартных блоков. Процесс программирования немного напоминает Lego Mindstorms EV3, и заключается

в выстраивании определенной последовательности пиктограмм с заданными параметрами.

На базе Scratch позже было создано несколько сред разработки,



использующих визуально-блочную, событийно-ориентированную схему программирования. Наиболее удачная из них, на мой взгляд, mBlock. Помимо простоты создания алгоритмов, характерных всем Scratch-подобным средам, эта среда включает поддержку очень большого количества контроллеров, драйвера для которых при необходимости легко подгружаются из сети Internet. Кроме того, при помощи mBlock можно создавать, открывать и редактировать файлы Scratch. Так что, используя данную среду, учащиеся могут не только освоить программирование контроллеров и микрокомпьютеров широкой номенклатуры, но и создавать анимацию, игры и приложения.

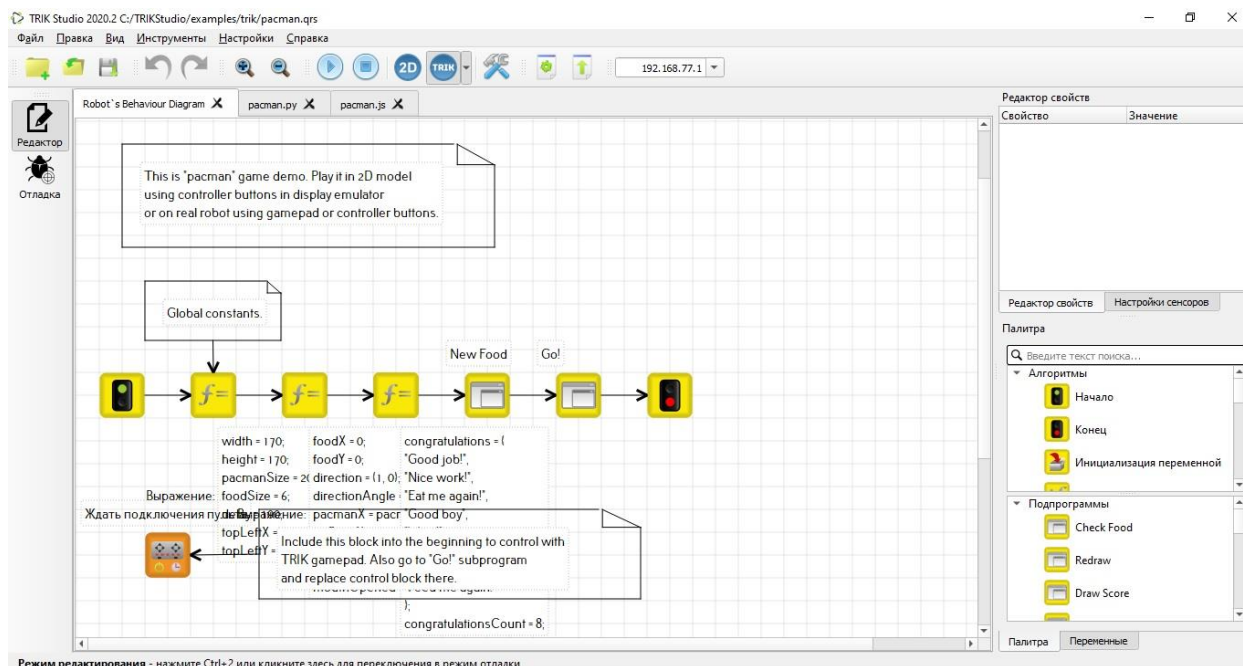


Дополнительным бонусом к программированию в среде mBlock является возможность написания кода на языке Python, что дает возможность попробовать себя во «взрослом», консольном программировании.

#### 4. Среда TRIK Studio.

Следующая среда, на которой я хочу остановиться, это TRIK Studio. Данный продукт разработан в России, входит в реестр отечественного ПО и распространяется бесплатно. По сравнению с рассмотренными ранее программами, это более продвинутый, но и более сложный в освоении инструмент. Как и в mBlock, тут присутствует среда визуального программирования и консольная среда. Визуальная среда в TRIK Studio не походит ни на одну среду, описанную выше, имеет другую идеологию и напоминает скорее промышленную визуализированную систему программирования (мне, например, напомнила STEP 7 от компании Siemens). Консольная же среда позволяет писать код не только на Python, но и на

## JavaScript.



Еще одна отличительная черта данного продукта – его универсальность. TRIK Studio позволяет программировать не только EV3 и NXT от Lego, но и собственные конструкторы TRIK (для чего среда и создавалась изначально), а также некоторые виды коптеров.

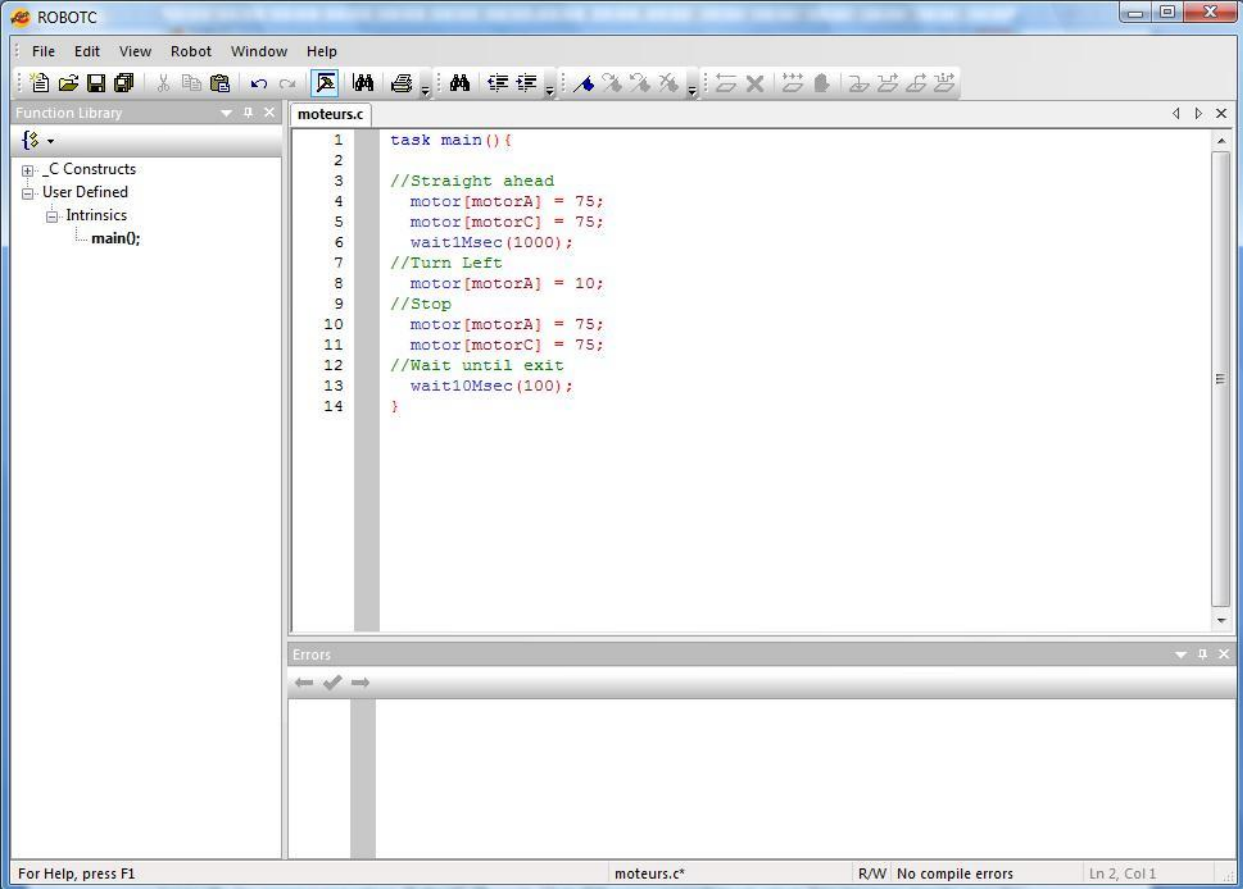
Ну и главная особенность этого программного продукта заключается в следующем: для того чтобы увидеть правильно или нет работает программа, загруженная в робота, не обязательно иметь под рукой самого робота. В TRIK Studio есть режим визуализации работы конечного устройства под управлением созданной программы.

На мой взгляд, все вышеописанное делает среду TRIK Studio оптимальной для освоения робототехники на начальном этапе.

### 5. Среда RobotC.

RobotC - самая сложная среда программирования роботов из моего обзора. Тут можно забыть про визуальное программирование, только текст,

ТОЛЬКО КОНСОЛЬ. Пример окна среды ниже.



```
1  task main() {
2
3  //Straight ahead
4  motor[motorA] = 75;
5  motor[motorC] = 75;
6  wait1Msec(1000);
7  //Turn Left
8  motor[motorA] = 10;
9  //Stop
10 motor[motorA] = 75;
11 motor[motorC] = 75;
12 //Wait until exit
13 wait10Msec(100);
14 }
```

Разработкой данной платформы занималась компания Lego в содружестве с Carnegie Mellon. Продукт предназначается для более углубленного изучения робототехники и программирования.

Основой RobotC, как видно из названия, послужил язык программирования C. Отсюда вытекают все плюсы и минусы среды.

К плюсам нужно отнести универсальность (можно создавать код для очень большого количества контроллеров различных производителей), а также чрезвычайную гибкость (так как основан на языке C).

К минусам же относится сложность освоения. Тут ни о каком «нулевом» уровне в программировании речи, конечно не идет. Для освоения данной



среды необходимы некоторые познания в программировании (желательно на языках низкого уровня), его основных понятиях и терминах.

### **Заключение.**

Вот, пожалуй, и все, о чем я хотел написать в данном обзоре. Конечно, существует еще масса программных сред, кроме описанных выше, позволяющих работать с компьютером такого замечательного конструктора, как Lego Mindstorms EV3. Они имеют разный подход к программированию, требуют различного уровня подготовки. Но все эти продукты являются эффективными инструментами при работе с EV3, и позволяют развивать уровень компетенции в области образовательной робототехники. Так что среди этого многообразия каждый найдет подходящую ему систему.

А.Е. Степанов, педагог дополнительного образования  
МБУ ДО «ДДТ», ГО Среднеуральск, 2023 г.