

Областное бюджетное учреждение дополнительного образования
«Областной центр развития творчества детей и юношества»

«Основные параметры установки и команды в VEX IQ»
Методические рекомендации

Автор:
Пикалов Владимир Анатольевич,
педагог дополнительного образования
первой квалификационной категории

г. Курск, 2023 г.

Данная методическая разработка составлена с целью оказания помощи молодым педагогам, которые только начинают работу с образовательным конструктором VEX IQ, создание условий для эффективной организации, координации, управления проведением занятия по робототехнике, повышения профессиональной компетентности педагогического работника по направлению «Робототехника».

Основные параметры установки и команды в VEX IQ

План:

1. Что такое визуальное (графическое) программирование?
2. Как установить и откуда?
3. Основные блоки команд
4. Первая программа
5. Подключение робота
6. Запуск программы

Языки программирования появились еще в XIX веке. Работа с вычислительными машинами с помощью языков программирования требовала специальных сложных умений. Одновременно с этим вычислительные системы становились сложнее, и к моменту, когда появились первые компьютеры, языков программирования было много, они были разными и требовали долгого обучения работе с ними.

Проекты с использованием цифровых технологий разрастались – команды разработчиков состояли не только из программистов, но и ученых самых разных областей, каждый из которых разбирался только в своей отрасли. Проблема была в том, что судьба проекта сильно зависела от того, как точно все участники проекта поймут друг друга и смогут ли предложить наиболее эффективные решения. Ученые понимали, что необходимо средство, которое даст безусловное единое понимание алгоритмов для всех участников любого проекта. Одним из таких средств стали визуальные языки программирования.

Визуальный язык программирования позволяет вместо написания текста программы использовать графические объекты. Ранние визуальные языки были похожи на блок-схемы. Одним из ярких примеров можно назвать язык ДРАКОН, активно использовавшийся в 80-е годы XX века для разработки системы управления космического корабля «Буран». Другой пример – более современный LabVIEW, предназначенный для систем сбора и обработки данных.

Сегодня существует много графических сред для обучения программированию в начальной и средней школах. Визуальные языки программирования позволили снизить возраст детей, начинающих заниматься программированием. Программирование роботов стало популярным благодаря упрощенности работы с визуальными языками и наглядности потоков данных внутри программы. Безусловно, упрощенность языка играет и отрицательную роль, например в возможностях создания сложных устройств, но для развития устойчивого интереса к программированию этого не требуется.

Данное методическое пособие предлагает познакомиться с графической средой программирования VEXcode IQ blocks.

VEXcode IQ blocks старается совместить в себе плюсы графического программирования (наглядность потоков данных) и текстового программирования

(все основные операторы управления, различные виды циклов, работа с булевой логикой и массивами, обработка событий, сопровождение команд текстом на английском языке).

Для того чтобы установить среду программирования, необходимо скачать установочный пакет с официального сайта (Рисунок 1).



Рисунок 1. Страница скачивания среды программирования VEXcode IQ Blocks

Далее при запуске установочного файла остальной процесс происходит автоматически и не требует особенных умений.

При запуске программы сразу открывается новый файл для создания программы (Рисунок 2).

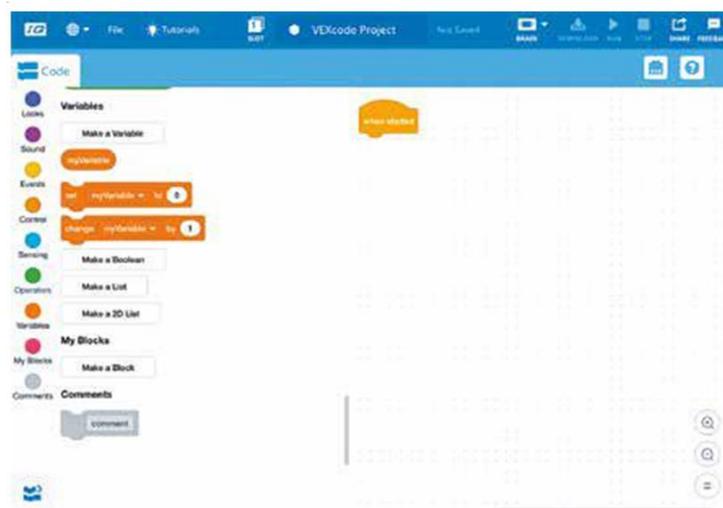
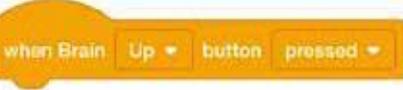
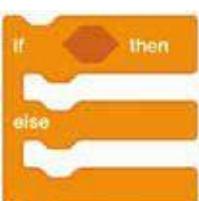


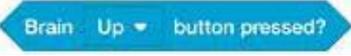
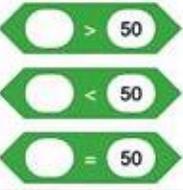
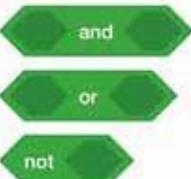
Рисунок 2. Файл для создания программы в среде VEXcode IQ Blocks

Все основные команды разделены на 9 категорий, подробное описание которых представлено в таблице 1.

Таблица 1

Раздел	Название	Команда	Описание
 Looks	Вывод на экран	Looks 	Вывод на экран
			Установить курсор на строку _ и столбец _
			Установить курсор на следующую строку
			Установить точность вывода
			Очистить строки
			Очистить строку _
 Sound	Звуки	Sound 	Воспроизвести звук
			Воспроизвести ноту
 Events	События	Events 	Начало программы
			Когда кнопка блока нажата/отпущена
			Когда таймер больше _ секунд
			Когда получено сообщение
			Транслировать сообщение

			Транслировать сообщение и ждать
	Управление	Control 	Ждать заданное количество секунд
			Повторять заданное количество раз
			Повторять бесконечно
			Если <условие>, то ...
			Если <условие>, то ... , иначе ...
			Ждать до тех пор, пока не произойдет заданное условие
			Повторять до тех пор, пока не произойдет заданное условие
			Пока <условие> верно, повторять
			Прерывание цикла

Раздел	Название	Команда	Описание
 Sensing	Измерение	Sensing 	Сбросить таймер
			Установить таймер в секундах
			Столбец курсора
			Строка курсора
			Нажата ли кнопка?
			Процент заряда батареи
 Operators	Операторы	Operators    	Операторы сложения, вычитания, умножения и деления
			Генерация случайного числа в заданном диапазоне
			Операторы сравнения: больше, меньше, равно
			Логические операции: И, ИЛИ, НЕ
			Округление

Раздел	Название	Команда	Описание
			<p>Функции:</p> <ul style="list-style-type: none"> - получение модуля, - округление вниз, - округление вверх, - квадратный корень, - синус, - косинус, - тангенс, - котангенс, - арксинус, - арккосинус, - арктангенс, - десятичный логарифм, - логарифм, - e^x, - 10^x
			Остаток от деления
 Variables	Переменные	<p>Variables</p> <p>Make a Variable</p> <p>myVariable</p>	Создать переменную
			Установить значение переменной
			Изменить значение переменной
		Make a Boolean	Создать переменную логического типа
		Make a List	Создать список
		Make a 2D List	Создать двухмерный список
 My Blocks	Мои блоки	<p>My Blocks</p> <p>Make a Block</p>	Создать собственную функцию
	Комментарии		Создать комментарий



Блоки других функциональных групп (например, блоки **Движения (Motion)**) будут подробно рассмотрены в других занятиях. Они появляются при подключении приводов и других устройств.

Составим первую программу вывода на экран приветствия и воспроизведения звука. Для этого после блока **When Started** вытащим и присоединим команду **Print** со значением «Привет!» (Рисунок 3).

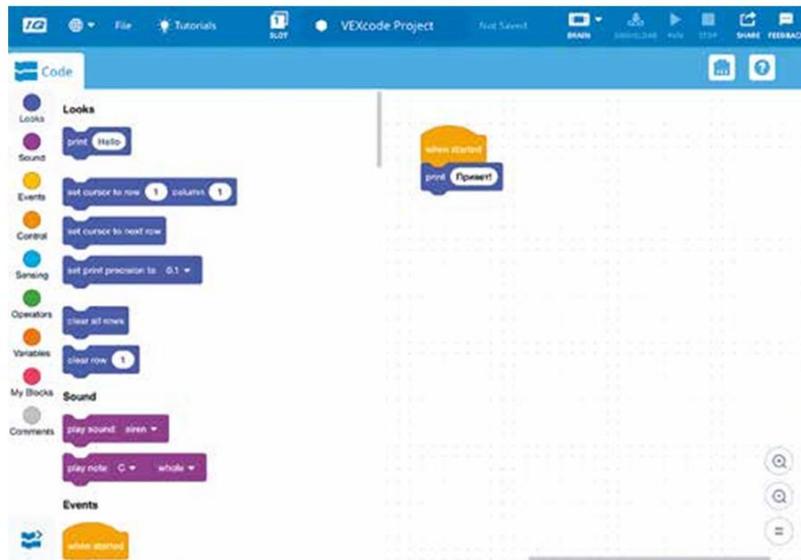


Рисунок 3. Вывод на экран слова «Привет!» с помощью команды **Print**

Затем выберем команду **Play Sound** и откроем выпадающий список (Рисунок 4).

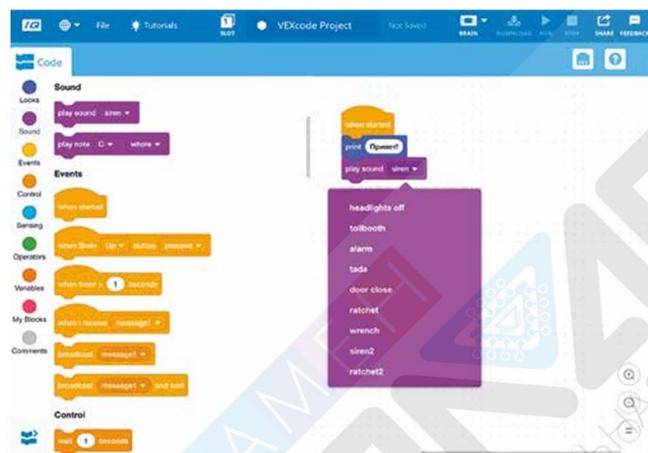


Рисунок 4. Выбор звука для воспроизведения

И после этого определимся, какой звук больше подходит, например «**tada**» (Рисунок 5).

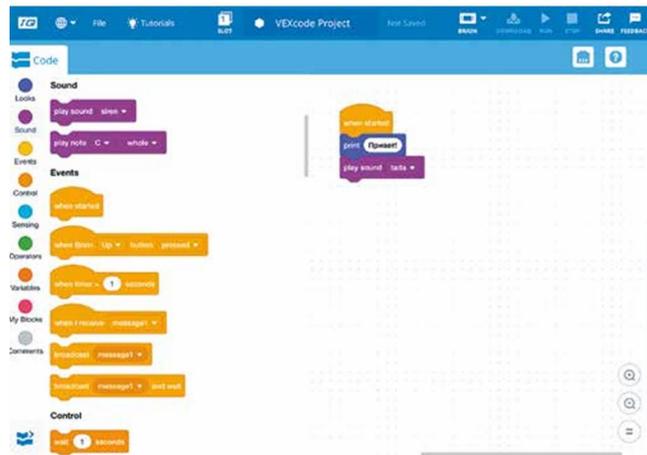


Рисунок 5. Выбор звука «tada» для воспроизведения

Программа может быть запущена после подключения блока к компьютеру или планшету. Подробные инструкции по подключению представлены в приложении к данному занятию. После того как блок VEX IQ подключен, значок **BRAIN** будет гореть зеленым цветом (Рисунок 6). После этого достаточно нажать кнопку **RUN** и программа будет запущена.

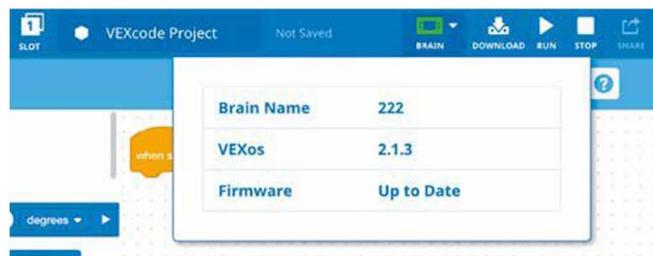


Рисунок 6. Индикатор подключения блока VEX IQ

Программу можно также загрузить в блок VEX IQ. Для этого достаточно нажать кнопку **DOWNLOAD**. После этого блок можно будет отсоединить от компьютера или планшета и использовать автономно.

Список используемой литературы

1. Иванов А.А. Основы робототехники. – М.: Форум, 2012. – 224 с.
2. Крейг Д. Введение в робототехнику. Механика и управление. – М.: Институт компьютерных исследований, 2013. – 564 с.
3. Филиппов С.А. Уроки робототехники. Конструкция. Движение. Управление. – М.: Лаборатория знаний, 2017. – 176 с.