

Проект «Математический робот-тренажер для дошкольников»

Левкович Иван Артемович

ГУО «Гимназия №13 г. Минска»

3 «Б» класс

Введение.

Моя младшая сестренка Василиса в следующем году станет первоклассницей. И я, как старший брат, решил помочь ей с подготовкой к школе. Поскольку математика и робототехника мои любимые предметы, я построю для нее робота, который в доступной и интересной форме научит ее считать.

Цель проекта.

Создание печатающего робота для обучения детей дошкольного возраста распознаванию цифр и решению математических примеров на сложение и вычитание в пределах девяти.

Задачи проекта.

1. Сборка робота из набора Lego Mindstorms EV3.
2. Создание таблицы символов цифр.
3. Программирование робота, включающее создание подпрограмм для:
 - генерирования случайных цифр и примеров;
 - печати цифр и математических символов;
 - распознавания ответов;
 - проверки ответов.
4. Тестирование робота и отладка работы программы.
5. Практическое использование робота-тренажера.

Сборка робота.

Lego Mindstorms EV3 – робототехнический набор для детского конструирования. В базовый набор входят: программируемый управляющий модуль, различные моторы и сенсоры, соединительные кабели, классические пластиковые детали Lego. С помощью этого набора удобно создавать робота в домашних условиях. Прежде чем приступить к работе, я изучил существующие схемы сборки в интернете, выбрал наиболее подходящую и по ней собрал своего робота.



Рисунок 1. Набор Lego Midstorms EV3

Компоненты робота:

1. Мотор А – механизм, управляющий движением фломастера по горизонтали.
2. Мотор В – механизм, управляющий подачей бумаги.
3. Мотор С – механизм, управляющий положением фломастера над бумагой.
4. Цветовой сенсор, определяющий наличие бумаги в принтере.
5. Датчик касания - для ввода и декодирования ответов.
6. Программируемый управляющий модуль.

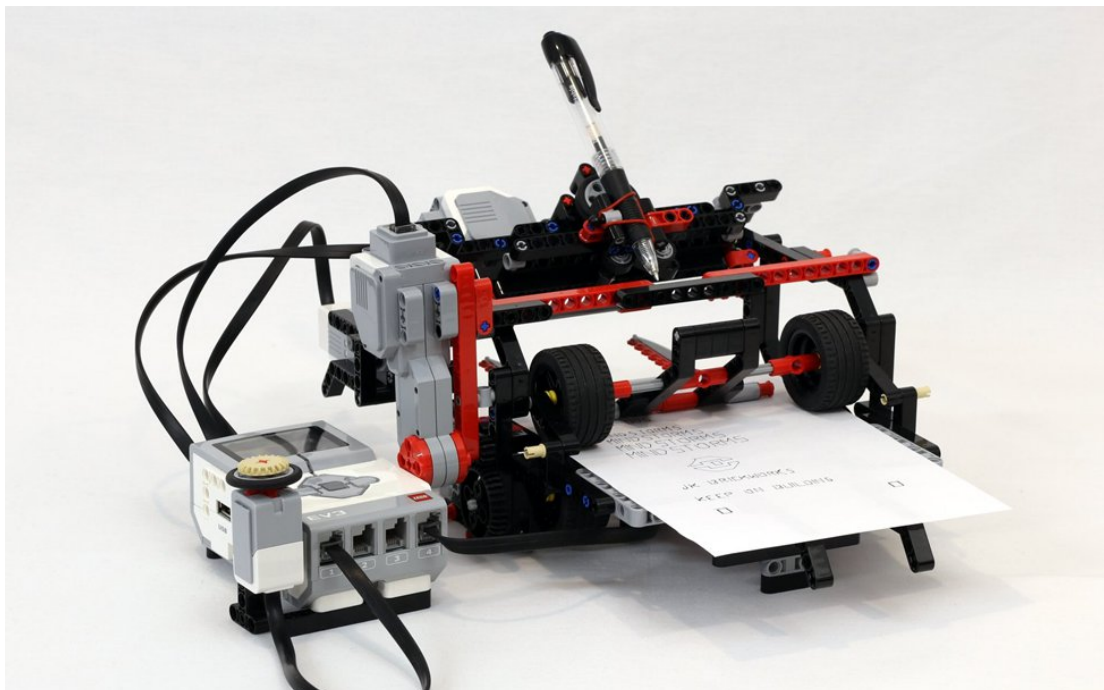


Рисунок 2. Печатающий робот тренажер

Создание таблицы символов цифр.

Образец написания символов цифр я взял с почтового конверта. Перенес цифры в тетрадь в клеточку и принялся за программирование.



Рисунок 3. Примеры написания символов цифр

Программирование робота.

По моей задумке для тренировки математических навыков робот будет предлагать 2 вида упражнений:

1. распознавание цифр
2. решение математических примеров на сложение и вычитание в пределах девяти

Количество предлагаемых цифр и примеров может изменяться в настройках программы.

В среде EV3 используется графический язык программирования, в котором команды представлены в виде блоков. Готовые блоки можно перетаскивать и соединять между собой в необходимой последовательности.



Рисунок 4. Пример записи алгоритма на языке программирования EV3

1. **Алгоритм для генерирования случайных примеров и чисел.** В стандартное программное обеспечение модуля Lego Mindstorms EV3 входит

программный блок случайных событий, который я использовал для генерирования случайных цифр.

Для того чтобы сгенерировать случайный пример я придумал следующий алгоритм:

- генерировал первое случайное однозначное число;
- генерировал случайную операцию (сложение или вычитание);
- в зависимости от операции генерировал второе однозначное число с условием, если операция сложение, то второе число должно быть не больше, чем разность девяти и первого числа (чтобы сумма слагаемых не получилась больше 9), если операция вычитание, то второе число должно быть не больше, чем первое число (чтобы в результате не получилось отрицательное число).

Далее с помощью алгоритма печати сгенерированные цифры и примеры поочередно печатаются на бумаге.

2. **Алгоритм для печати цифр и математических символов.** Для управления движением фломастера я опытным путем определил, что поворот мотора А на 9 градусов приводит к написанию горизонтальной линии длиной 1 мм, поворот мотора В на 9 градусов – вертикальной линии длиной 1 мм, одновременный поворот моторов А и В на 9 градусов – диагональной линии длиной около 1.4 мм. Далее я создал отдельные подпрограммы для написания горизонтальных, вертикальных и диагональных линий нужной мне длины, а из готовых линий программировал цифры.
3. **Алгоритм для распознавания ответов.** Для распознавания ответов я использовал датчик нажатия, заложив простой принцип: число нажатий соответствует аналогичной цифре.
4. **Алгоритм проверки ответов.** Для проверки правильности распознавания цифр программа сравнивает число нажатий на датчик нажатия с напечатанной цифрой. При правильном ответе управляющий модуль воспроизводит сигнал «аплодисменты» и переходит к генерированию и печати следующей цифры. При неправильном ответе модуль воспроизводит сигнал «о-оу» и предоставляется повторная возможность для ввода ответа. Далее вновь следует проверка ответа и так продолжается до тех пор, пока не будет дан правильный ответ. После генерации шести случайных цифр, программа подсчитывает число неправильных ответов и печатает результат выполнения задания. Для проверки правильности решения случайных примеров программа вычисляет правильный ответ самостоятельно и сравнивает с числом нажатий на датчик нажатия. Проверка правильности решения проводится так же, как и в случае с распознаванием цифр: до правильного ответа. При правильном ответе соответствующая цифра выводится в печать. После генерации трех случайных примеров программа подсчитывает число неправильных ответов и печатает его.

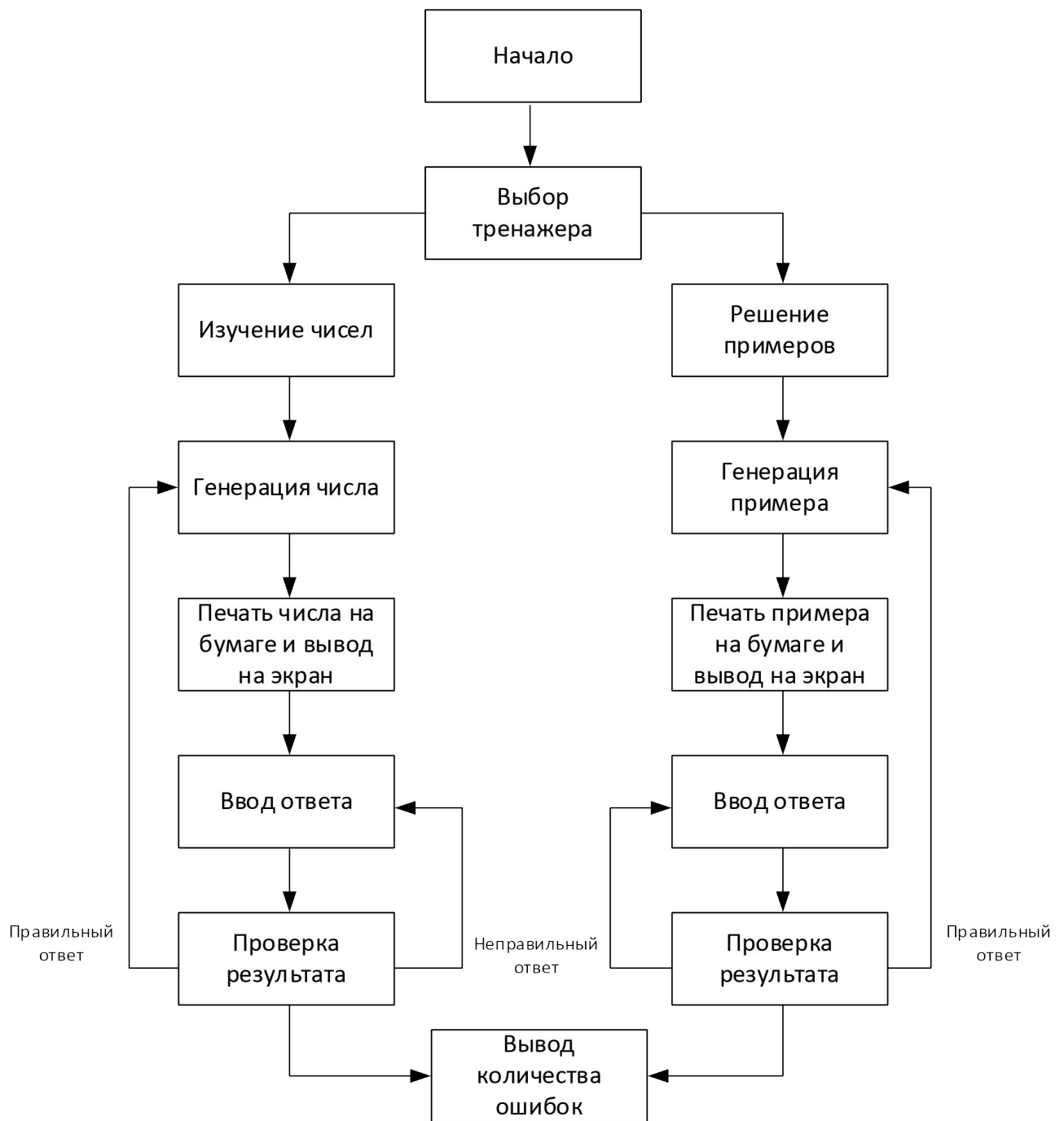


Рисунок 5. Блок-схема работы робота-тренажера

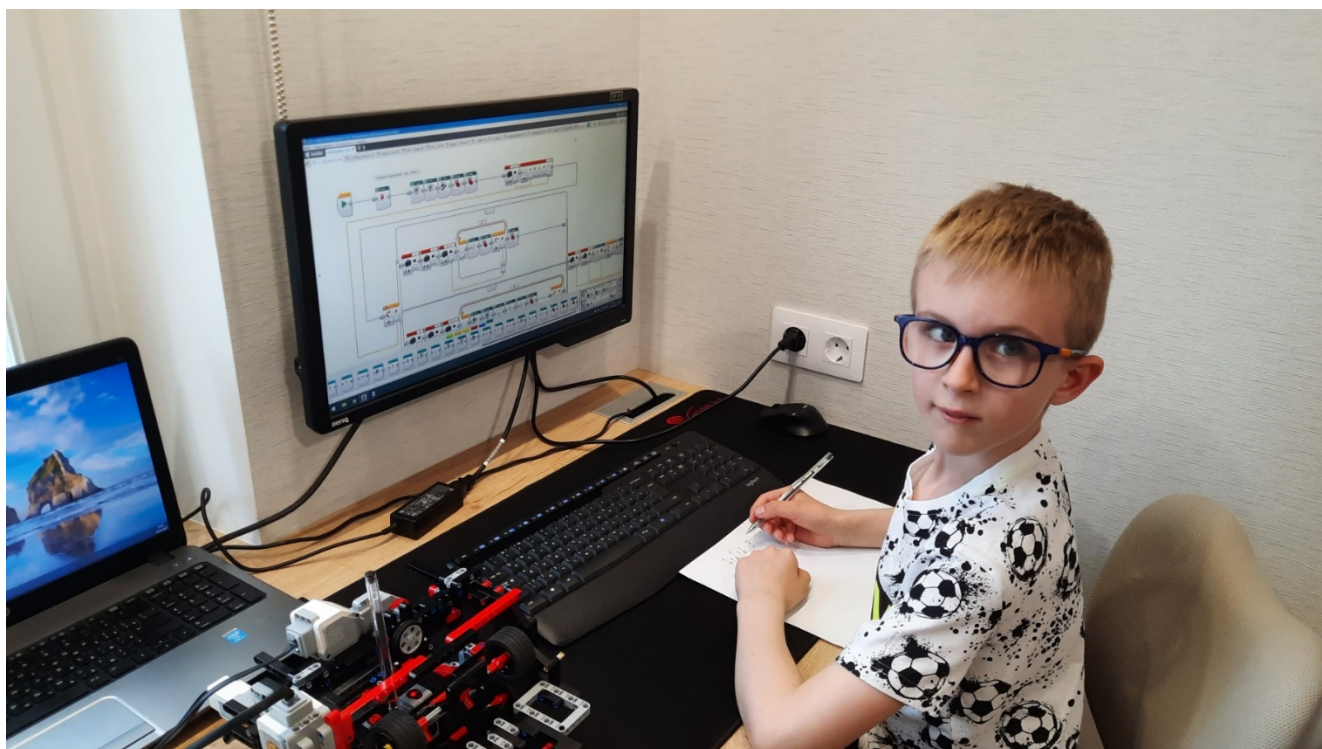
Тестирование робота и отладка работы программы.

Программируя своего робота, я столкнулся с определенными трудностями, как с механической частью робота, так и с его программированием.

Например, использование трения резинового колеса о балку для перемещения ручки, а также люфты и зазоры между зубьями зубчатых передач, не позволяют добиться требуемой точности к длине линий, которые рисуются на листке бумаги. Для исправления этой погрешности в некоторых случаях мне приходилось дополнительно увеличивать градус вращения мотора

В процессе написания программы я понял, что оригинальная среда разработки EV3 не подходит для написания сложных и многомодульных программ. Из-за использования графических блоков, программа получается большой, ее неудобно анализировать и исправлять ошибки. Сложные алгоритмы с трудом помещаются на экран монитора. Для уменьшения размера основного алгоритма мне приходилось разбивать основную программу на подпрограммы и создавать отдельные модули.

Так же эта среда не поддерживает русский язык, поэтому названия подпрограмм и интерфейс управляющего блока написаны на английском языке. Для перевода я использовал Google translate.



Практическое использование тренажера.

С этой задачей я пока не справился на 100%. Но процесс обучения сестры уже запущен. На данный момент наши уроки приносят ей только удовольствие. Обучение проходит в непринужденной, игровой форме. Я тоже очень рад на минуточку почувствовать себя учителем. Думаю, в ближайшее время мы освоим цифры и научимся считать, а в дальнейшем я построю нового робота, который научит ее читать.



Конечный результат.

«Если очень захотеть, можно в космос полететь!»

Мне удалось построить и запрограммировать печатающего робота-помощника, который поможет подготовить мою сестренку к школе.

В дальнейшем я планирую усовершенствовать моего робота следующим образом:

- использовать зубчатые передачи и зубчатую рейку для преобразования вращательного движения мотора в горизонтальное движение ручки, чтобы добиться точности написания цифр;
- добавить в программу таймер, который позволит определить затраченное время на выполнение задания;
- ввести систему оценки выполненного задания, учитывая правильность решения и скорость;
- повысить сложность случайно генерируемых примеров, добавить функции умножения и деления.

Так же я планирую переписать программу, используя язык программирования Python, который позволяет программировать модуль EV3 не графическими блоками, а текстовыми командами. Это позволит мне программировать более сложные задачи.