Научная статья
УДК 372.800.2
DOI: 10.24412/2072-9014-2026-276-41-49

ДЕКОМПОЗИЦИЯ КОМАНД УПРАВЛЕНИЯ МОТОРАМИ ИЗ ОБРАЗОВАТЕЛЬНЫХ НАБОРОВ ПО РОБОТОТЕХНИКЕ В ВИДЕ БЛОК-СХЕМ ДЛЯ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ РОБОТОТЕХНИЧЕСКИХ ПЛАТФОРМ

Игнатий Игнатьевич Мацаль

Московский городской педагогический университет,
Москва, Россия
ignatmatsal@mail.ru, https://orcid.org/0009-0006-3325-7162

Аннотация. В статье представлен алгоритмический подход к декомпозиции системы управления моторами в образовательных наборах по робототехнике. Рассматривается структуризация блоков по параметрам: базовые, входные и выходные. Компоновка параметров между собой позволяет формировать элементарные модули для интеграции полученных блок-схем в процесс изучения программирования выбранной образовательной робототехнической платформы. Проведен теоретический анализ, показаны примеры использования, приведен пример урока для школьных педагогов.

Ключевые слова: робототехника; моторы; образовательные наборы; информатика; блок-схемы.

Для цитирования: Мацаль И. И. Декомпозиция команд управления моторами из образовательных наборов по робототехнике в виде блок-схем для обучения программированию робототехнических платформ / И. И. Мацаль // Вестник МГПУ. Серия «Информатика и информатизация образования». 2026. № 2 (76). С. 41–49. https://doi.org/10.24412/2072-9014-2026-276-41-49

Scientific article
UDC 372.800.2
DOI: 10.24412/2072-9014-2026-276-41-49

DECOMPOSITION OF MOTOR CONTROL COMMANDS FROM EDUCATIONAL ROBOTICS KITS INTO FLOWCHARTS FOR TEACHING PROGRAMMING OF ROBOTICS PLATFORMS

Ignatiy I. Matsal

Moscow City University,
Moscow, Russia
ignatmatsal@mail.ru, https://orcid.org/0009-0006-3325-7162

Abstract. The article presents an algorithmic approach to the decomposition of the motor control system in educational robotics kits. The structuring of blocks based on parameters — basic, input, and output — is examined. Combining these parameters allows for the formation of elementary modules to integrate the resulting flowcharts into the process of learning the programming of a selected educational robotics platform. A theoretical analysis is conducted, usage examples are demonstrated, and a sample lesson for schoolteachers is provided.

Keywords: robotics; motors; educational kits; computer science; flowcharts.

For citation: Matsal I. I. Decomposition of motor control commands from educational robotics kits into flowcharts for teaching programming of robotics platforms / I. I. Matsal // MCU Journal of Informatics and Informatization of Education. 2026. № 2 (76). P. 41–49. https://doi.org/10.24412/2072-9014-2026-276-41-49

Введение

Рост интереса к изучению робототехники в образовательных учреждениях [1; 2] требует совершенствования методов преподавания основ программирования робототехнических образовательных платформ. В настоящее время существует множество образовательных решений, которые не только позволяют изучать языки программирования в рамках выбранной платформы, но и формируют алгоритмическое мышление у обучающихся [3].

Разнообразие платформ ведет к тому, что существующие блок-схемы при составлении алгоритмов работы робототехнических систем становятся слишком абстрактными и не позволяют быстро перейти от визуального представления к формированию кода (независимо от типа кода: блочно-модульное представление или текстовый язык) [4]. На рисунке 1 представлена классическая блок-схема для запуска электрического привода.

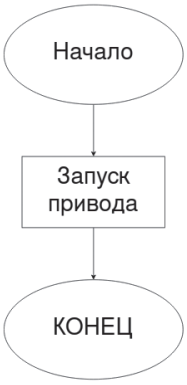

Рис. 1. Блок-схема для запуска электрического привода

Данная блок-схема описывает общий принцип работы робототехнической системы, в рамках которой необходимо запустить электрический привод, но она не позволяет понять первоначальные настройки электрического привода и более детальные параметры, которые помогут учащемуся получить предварительную блок-схему для моделирования программы под выбранную среду программирования и платформу. Кроме того, более детальные параметры помогут сформировать представление об устройствах как о сложном комплексе настроек и параметров, что позволит минимизировать число ошибок при написании кода и получить большую компетенцию.

В данной статье будут представлены анализ команд управления моторами в средах разработки *Arduino IDE, VEXcode* и *Lego Mindstorm EV3*, а также разбиение на элементарные блоки для составления алгоритмов работы моторов.

1. Сравнение команд управления моторов в средах разработки Arduino IDE, VEXcode и Lego Mindstorm EV3

Одним из ключевых элементов образовательных робототехнических платформ является система управления устройствами, реализованная через блочное или текстовое программирование. Необходимость поиска путей рационализации структуры управления устройствами приводит к разработке простых и наглядных средств перехода от алгоритмов к конкретным программным решениям [5] через конкретизацию параметров управления, представленных в виде элементов блок-схем.

Для дальнейшего исследования необходимо рассмотреть особенности выбранных сред программирования посредством изучения команд (методов и функций) управления моторов, используя базовые примеры кода. В качестве базовых примеров будет рассмотрен запуск электрического привода для выбранных сред разработки.

- *Arduino IDE.*

Листинг 1

Пример кода с комментариями для запуска привода в среде разработки *Arduino IDE*

```
// Мотор подключен через драйвер к PWM-пину 9 (например, через L298N).
// Описывает, что используем пин 9 для управления мотором:
const int motorPin = 9;
void setup() {
  // Переводим пин 9 в режим выхода (OUTPUT), чтобы отправлять на него
     управляющие сигналы:
  pinMode(motorPin, OUTPUT);
}

void loop() {
  // Отправляем на пин 9 PWM-сигнал уровня 200 (из возможных 255), что
     эквивалентно примерно 78 % мощности мотора:
  analogWrite(motorPin, 200);
}
```

Разъяснения:

– *const int motorPin* = 9 — определяем номер пина для подключения мотора;

– *pinMode(motorPin, OUTPUT)* — назначаем пин для вывода сигнала;

– *analogWrite(motorPin, 200)* — отправляем на мотор *PWM*-сигнал определенного уровня (скорость работы мотора).

- *VEXcode V5 (C++).*

Листинг 2

Пример кода с комментариями для запуска привода в среде разработки *VEXcode V5*

```
// Подключение основной библиотеки VEXcode API:
#include «vex.h»

// Создание объекта Brain — основной контроллер:
vex::brain Brain;

// Создаем объект Motor1, который подключен к 1-му порту модуля:
vex::motor Motor1(vex::PORT1);

int main() {
  // Запускаем Motor1 вперед с 80% мощности (pct):
  Motor1.spin(vex::directionType::fwd, 80, vex::velocityUnits::pct);
  while(true) {
    // Оставляем мотор включенным.
    // Короткая пауза (100 милисекунд), чтобы программа не завершалась:
    vex::task::sleep(100);
  }
}
```

Разъяснения:

– *vex::motor Motor1(vex::PORT1)* — создаем объект мотора, который подключен к порту 1;

– *Motor1.spin*(...) — запускаем мотор вперед с заданной скоростью (80 %).

• *LEGO Mindstorms EV3 (MicroPython).*

Листинг 3

Пример кода с комментариями для запуска привода в среде разработки LEGO Mindstorms EV3 (MicroPython)

```
# Импорт класса Motor для работы с моторами EV3:
from pybricks.ev3devices import Motor
# Импорт Port для работы с номерами портов:
from pybricks.parameters import Port
# Создаем объект 'motor' и привязываем его к порту A:
motor = Motor(Port.A)
# Запуск мотора на скорости 360 градусов/сек. Мотор крутится до команды stop():
motor.run(360)
```

Разъяснения:

– *from pybricks.ev3devices import Motor* — импортируем класс управления моторами EV3;

– *from pybricks.parameters import Port* — импортируем перечисление с именами портов (*Port.A, Port.B*, и т. д.);

– *motor = Motor(Port.A*) — инициализируем мотор на порту A, создаем объект для работы с мотором;

– *motor.run(360)* — передаем мотору команду крутиться на скорости 360 градусов/секунду (стандартное значение для *EV3*). Мотор будет работать, пока не поступит другая команда (например, *stop*()).

Результаты сравнения:

• везде требуется обозначить тип устройства, порт (пин) подключения устройства и задать скорость устройства;

• в коде Arduino основной цикл управления выполняется бесконечно — используется низкоуровневое задание сигнала;

• на VEX и EV3 используются более высокоуровневые команды и объекты, соответствующие моторам.

2. *Представление команд управления моторов в средах разработки Arduino IDE, VEXcode и Lego Mindstorm EV3 в виде декомпозированных блоков*

В результате анализа фрагментов кода можно сделать вывод о том, что среды программирования имеют схожие алгоритмы работы системы управления

мотором, что позволяет нам разделить выделенные параметры на три группы параметров (рис. 2).

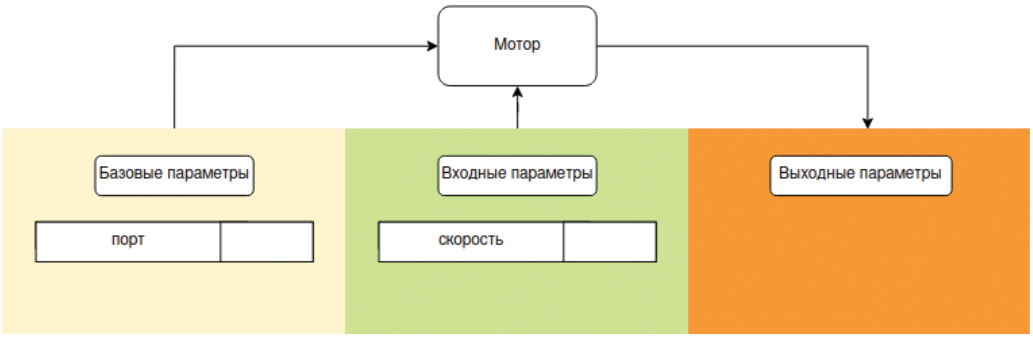

Рис. 2. Декомпозиция системы управления мотором в виде блоков

Компоновка данных блоков дает в базовом представлении объект мотора, который можно включить в классическую блок-схему при составлении алгоритма работы привода (рис. 3).

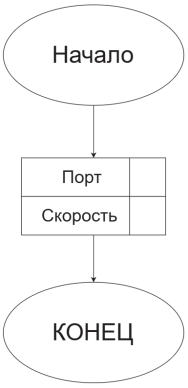

Рис. 3. Алгоритм работы привода в виде блок-схем
с использованием декомпозированных блоков

3. Методическая ценность декомпозиции команд управления для преподавания робототехники и алгоритмизации в школе

Предлагаемый метод декомпозиции команд управления моторами обладает высокой практической значимостью для школьных педагогов информатики и технологии, сталкивающихся со сложностями перехода от наглядно-образного к абстрактно-логическому мышлению у учащихся. Основная методическая ценность подхода заключается в создании эффективного моста между классическим представлением алгоритмов в виде блок-схем и реальной практикой программирования платформ по образовательной робототехнике (текстовый код на языках *C++*, *Python*).

Во-первых, предлагаемая методика позволяет устранить эффект черного ящика. Как правило, при работе с визуальными языками программирования обучающийся использует готовую команду запуска двигателя, не вникая в физику процесса, в то время как данный подход требует от школьника глубокой детализации алгоритма: самостоятельного назначения порта подключения, определения рабочего режима и уровня мощности. В результате аппаратное обеспечение трансформируется из скрытой структуры в прозрачную параметрическую модель. Подобная практика стимулирует развитие инженерного мышления, что является критически важным аспектом для освоения робототехники.

Во-вторых, рассматриваемая технология служит эффективной базой для последующего перехода к текстовому программированию. Предварительное знакомство со структурированными алгоритмами в блочном виде существенно облегчает миграцию на Python или C++ (включая платформы Arduino и VEXcode) [6]. Приступая к написанию скриптов, учащийся осознает внутреннюю логику программы, а не занимается механическим дублированием шаблонных примеров. Это минимизирует вероятность возникновения типовых синтаксических недочетов, таких как пропуск аргументов или отсутствие инициализации пинов, позволяя перенести фокус внимания на алгоритмическую оптимизацию.

В-третьих, метод отличается высокой степенью универсальности и кроссплатформенностью, поскольку преподаватель обеспечивается инструментарием, не привязанным к конкретной элементной базе. Декомпозиционный подход сохраняет свою актуальность вне зависимости от применяемой среды разработки (будь то LEGO, VEX или Arduino). Следовательно, образовательный процесс выстраивается вокруг фундаментальных законов управления киберфизическими устройствами, а не сводится к изучению нюансов отдельного конструктора. Эта концепция полностью отвечает целевым установкам ФГОС в части создания целостной информационной картины мира [7] и дает возможность гибко адаптировать рабочие программы под любую материально-техническую базу учебного заведения.

Внедрение детализированных декомпозированных алгоритмических схем способствует росту автономности школьников при освоении незнакомых платформ. Опираясь на развернутую схему действий, содержащую исчерпывающую информацию о входящих и исходящих переменных, учащиеся способны без постоянного контроля со стороны педагога проектировать нетривиальные алгоритмы перемещения роботов, а также осуществлять их отладку и корректировку. В итоге на занятиях высвобождается дополнительный временной ресурс, который можно направить на решение олимпиадных заданий и реализацию творческих проектов повышенной сложности.

Заключение

Резюмируя результаты проведенного исследования, можно утверждать: внедрение принципа декомпозиции в работу с управляющими блоками помогает

школьникам сформировать комплексное понимание того, как организуются и функционируют сложные исполнительные механизмы. Необходимость самостоятельно конструировать составные элементы команд повышает уровень концентрации внимания учащихся и ведет к более глубокому осмыслению принципов работы электродвигателей.

В качестве перспективного направления для будущих исследований рассматривается применение метода детализации систем управления к иным модулям образовательных комплексов. Кроме того, значительный научный потенциал имеет разработка процессов автоматизированной генерации программного кода непосредственно из визуальных блок-схем с опорой на выделенные параметрические данные.

Список источников

1. *Тарапата В. В.* Робототехника в школе: методика, программы, проекты / В. В. Тарапата, Н. Н. Самылкина. М.: Лаборатория знаний, 2026. 112 с.

2. A Systematic Review of Studies on Educational Robotics / S. Anwar [et al.] // Journal of Pre-College Engineering Education Research (J-PEER). 2019. Vol. 9. No. 2. Art. 2. DOI: 10.7771/2157-9288.1223

3. *Воробьева Н. А.* Развитие алгоритмического мышления у учащихся начальной школы с использованием системно-ориентированной среды программирования Scratch / Н. А. Воробьева // Вестник Пермского государственного гуманитарно-педагогического университета. Серия № 1. Психологические и педагогические науки. 2023. С. 59–65.

4. *Weintrop D.* Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms / D. Weintrop, U. Wilensky // ACM Transactions on Computing Education. 2017. No. 18 (1). P. 1–25. DOI:10.1145/3089799

5. *Абрамовских Н. В.* Методическое сопровождение педагогов начальной школы по реализации образовательных программ с применением робототехники / Н. В. Абрамовских, Е. С. Асланова // Вестник Сургутского государственного педагогического университета. 2021. № 4 (73). С. 61–69.

6. *Wang G.* From LEGO to Arduino: Enhancement of ECE Freshman Design with Practical Applications / G. Wang, J. Chen // 2016 ASEE Annual Conference & Exposition Proceedings. New Orleans: ASEE, 2016. DOI: 10.18260/p.26966

7. *Босова Л. Л.* Цифровые навыки современного школьника и возможности их формирования в школьном курсе информатики / Л. Л. Босова // Информатика в школе. 2020. № 7 (19). С. 5–9. DOI: 10.32517/2221-1993-2020-19-7-5-9

References

1. *Tarapata V. V.* Robotics in Schools: Methods, Programs, and Projects / V. V. Tarapata, N. N. Samylkina. Moscow: Knowledge Laboratory, 2026. 112 p.

2. A Systematic Review of Studies on Educational Robotics / S. Anwar [et al.] // Journal of Pre-College Engineering Education Research (J-PEER). 2019. Vol. 9. No. 2. Art. 2. DOI: 10.7771/2157-9288.1223

3. *Vorobyeva N. A.* The development of algorithmic thinking among elementary school students using the Scratch system-oriented programming environment / N. A. Vorobyeva //

Bulletin of the Perm State Humanitarian Pedagogical University. Series No. 1. Psychological and pedagogical sciences. 2023. P. 59–65.

4. *Weintrop D.* Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms / D. Weintrop, U. Wilensky // ACM Transactions on Computing Education. 2017. No. 18 (1). P. 1–25. DOI:10.1145/3089799

5. *Abramovskikh N. V.* Methodological support for primary school teachers on the implementation of educational programs using robotics / N. V. Abramovskikh, E. S. Aslanova // Bulletin of the Surgut State Pedagogical University. 2021. No. 4 (73). P. 61–69.

6. *Wang G.* From LEGO to Arduino: Enhancement of ECE Freshman Design with Practical Applications / G. Wang, J. Chen // 2016 ASEE Annual Conference & Exposition Proceedings. New Orleans: ASEE, 2016. DOI: 10.18260/p.26966

7. *Bosova L. L.* Digital skills of a modern student and the possibilities of their formation in the school computer science course / L. L. Bosova // Computer Science at school. 2020. No. 7 (19). P. 5–9. DOI: 10.32517/2221-1993-2020-19-7-5-9

Статья поступила в редакцию: 02.04.2026;
одобрена после рецензирования: 25.04.2026;
принята к публикации: 25.04.2026.

The article was submitted: 02.04.2026;
approved after reviewing: 25.04.2026;
accepted for publication: 25.04.2026.

Информация об авторе / Information about the author

Игнатий Игнатьевич Мацаль — аспирант, Институт цифрового образования, Московский городской педагогический университет, Москва, Россия.

Ignatiy I. Matsal — Postgraduate Student, Institute of Digital Education, Moscow City University, Moscow, Russia.

ignatmatsal@mail.ru, https://orcid.org/0009-0006-3325-7162