

2026, № 06 (июнь)

Раздел 5.8. Педагогика

ART 261172

DOI: 10.24412/2304-120X-2026-11172

УДК 372.862

Декомпозиция команд управления датчиками из образовательных наборов по робототехнике в виде блок-схем для обучения программированию робототехнических платформ

Decomposition of Sensor Commands from Educational Robotics Kits into Flowcharts for Teaching Robotics Platform Programming

Автор статьи

Мацаль Игнатий Игнатьевич,
аспирант ГАОУ ВО «Московский городской педагогический университет», г. Москва, Российская Федерация
ignatmatsal@mail.ru
ORCID: 0009-0006-3325-7162

Author of the article

Ignatiy I. Matsal,
Postgraduate Student, Moscow City Pedagogical University, Moscow, Russian Federation
ignatmatsal@mail.ru
ORCID: 0009-0006-3325-7162

Конфликт интересов

Конфликт интересов не указан

Conflict of interest statement

Conflict of interest is not declared

Для цитирования

Мацаль И. И. Декомпозиция команд управления датчиками из образовательных наборов по робототехнике в виде блок-схем для обучения программированию робототехнических платформ // Научно-методический электронный журнал «Концепт». – 2026. – № 06. – С. 548–560. – URL: <https://e-koncept.ru/2026/261172.htm> – DOI: 10.24412/2304-120X-2026-11172

For citation

I. I. Matsal, Decomposition of Sensor Commands from Educational Robotics Kits into Flowcharts for Teaching Robotics Platform Programming // Scientific-methodological electronic journal "Koncept". – 2026. – No. 06. – P. 548–560. – URL: <https://e-koncept.ru/2026/261172.htm> – DOI: 10.24412/2304-120X-2026-11172

Поступила в редакцию <i>Received</i>	25.02.26	Получена положительная рецензия <i>Received a positive review</i>	03.06.26
Принята к публикации <i>Accepted for publication</i>	03.06.26	Опубликована <i>Published</i>	30.06.26



Аннотация

Актуальность исследуемой проблемы обусловлена разнообразием робототехнических образовательных платформ в учебных заведениях и отсутствием единого методического подхода для изучения основ работы сенсорных систем. В образовательном процессе активно используются различные робототехнические конструкторы, обладающие уникальными архитектурами микроконтроллеров и специфическими средами программирования. Педагоги сталкиваются с серьезной проблемой отсутствия единого методического подхода к объяснению принципов работы сенсорных систем. Традиционные средства визуализации алгоритмов, такие как стандартные блок-схемы, представляют команды опроса датчиков как «черные ящики», полностью скрывая внутреннюю логику инициализации портов, выбора режима работы и формата обрабатываемых данных. Это формирует у учащихся исключительно поверхностное представление о сложных технических системах и создает существенные когнитивные барьеры при неизбежном переходе от визуального блочного программирования к написанию кода на языках программирования высокого уровня. Целью настоящего исследования является декомпозиция команд управления датчиками в виде блок-схем для обучения программированию робототехнических платформ для повышения эффективности обучения программированию роботов. Ведущим методом исследования выступил сравнительный анализ программных интерфейсов и синтаксических конструкций трех наиболее распространенных образовательных сред разработки. Кроме того, применялся метод структурного моделирования алгоритмов для выделения элементарных логических единиц управления периферийными устройствами. На основе этого вывода разработаны и представлены унифицированные графические блоки, позволяющие визуализировать алгоритм работы программы с глубокой детализацией параметров. Предложенный подход позволяет трансформировать абстрактную команду получения данных в четкую алгоритмическую структуру, полностью характеризующую работу физического устройства. Теоретическая значимость статьи состоит в научном обосновании необходимости внедрения этапа промежуточного моделирования, обеспечивающего плавный переход от наглядно-образного мышления к абстрактно-логическому через детализацию графических схем. Практическая значимость заключается в создании унифицированных графических блоков, позволяющих визуализировать логику работы датчиков до этапа написания программного кода.

Ключевые слова

образовательная робототехника, датчики, методика обучения программированию, алгоритмы, блок-схемы

Благодарности

Автор выражает благодарность ГАОУ ВО «Московский городской педагогический университет» за сотрудничество и предоставление доступа к необходимой информации.

Abstract

The relevance of the problem under study is driven by the diversity of educational robotics platforms in educational institutions and the lack of a unified methodological approach to teaching the fundamentals of sensor systems. Various educational robotics kits, featuring unique microcontroller architectures and specific programming environments, are actively utilized in the educational process. Educators face a significant challenge due to the lack of a unified methodological approach to explaining the operating principles of sensor systems. Traditional algorithm visualization tools, such as standard flowcharts, treat sensor polling commands as “black boxes,” completely obscuring the internal logic of port initialization, operating mode selection, and the format of the processed data. Consequently, students develop an exclusively superficial understanding of complex technical systems, which creates significant cognitive barriers during the inevitable transition from visual block-based programming to writing code in high-level programming languages. The objective of this research is to decompose sensor control commands into flowcharts for teaching robotic platform programming, thereby enhancing the overall effectiveness of robotics programming education. The primary research method was a comparative analysis of the software interfaces and syntactic structures of the three most widely used educational integrated development environments. Additionally, structural algorithm modeling was employed to identify the elementary logical units involved in peripheral device control. Based on these findings, unified graphical units have been developed and presented, enabling the visualization of the program’s algorithm with highly detailed parameters. The proposed approach transforms an abstract data retrieval command into a clear algorithmic structure that fully characterizes the operation of the physical device. The theoretical significance of this article lies in its scientific justification for introducing an intermediate modeling stage, which ensures a smooth transition from visual-figurative to abstract-logical thinking through the detailed elaboration of graphical diagrams. The practical significance lies in the development of unified graphical units that make it possible to visualize the logic of sensor operation prior to the stage of writing program code.

Key words

educational robotics, sensors, programming teaching methodology, algorithms, flowcharts

Acknowledgements

The author expresses gratitude to the Moscow State Pedagogical University for their cooperation and providing access to the necessary information.

Введение / Introduction

Рост интереса к изучению робототехники в образовательных учреждениях, отмечаемый в работах А. В. Параскевова, требует совершенствования методов преподавания основ программирования [1]. Данную тенденцию подтверждает и Т. В. Никитина, указывая на робототехнику как на ключевое направление инженерно-технического творчества школьников [2]. Современные образовательные стандарты смещают

фокус на концепцию «вычислительного мышления» (Computational Thinking), фундаментальную важность которой обосновывает Е. К. Хеннер [3]. В свою очередь, С. Лай детально рассматривает перспективы интеграции данного типа мышления через программирование в рамках школьных учебных программ [4]. Базовые основы этого феномена, ставшего фундаментом для многих образовательных инноваций, были изначально сформулированы в классической работе Дж. Винг [5]. Датчики представляют собой устройства различной степени сложности в зависимости от типа обрабатываемых данных и позволяют сделать мобильные или статические робототехнические платформы более интеллектуальными, а алгоритмы работы более сложными. В настоящее время существует множество образовательных решений, формирующих алгоритмическое мышление. Однако разнообразие платформ ведет к тому, что существующие блок-схемы становятся слишком абстрактными и не позволяющими конкретизировать настройку датчиков. Классический блок «Получить данные с датчика» в учебных материалах часто представляет собой «черный ящик», скрывающий процессы инициализации порта, настройки режима работы и обработки типа данных. Это не позволяет учащимся быстро перейти от визуального представления к формированию корректного текстового кода или блочно-модульного представления. Целью настоящего исследования является декомпозиция команд управления датчиками в виде блок-схем для обучения программированию робототехнических платформ. Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести сравнительный анализ синтаксических конструкций и интерфейсов программирования приложений (Application Programming Interface, далее – API) интегрированных сред разработки (Integrated Development Environment, далее – IDE) Arduino (C++), VEXcode V5 (C++) и LEGO Mindstorms EV3 (MicroPython) в контексте работы с периферийными устройствами.
2. Выявить общие паттерны инициализации и опроса сенсоров, независимые от специфики конкретного робототехнического конструктора.
3. Разработать унифицированные графические примитивы (блоки), визуализирующие логическую структуру команд управления с детализацией параметров (порт, режим, тип данных).
4. Обосновать методическую целесообразность применения разработанного подхода как средства снижения когнитивной нагрузки при переходе от визуального программирования к текстовому.

Обзор литературы / Literature review

Анализ современного состояния методики показывает, что ключевой проблемой остается формирование у учащихся глубокого понимания взаимосвязи между программным кодом и физическими процессами. В российской педагогической науке Л. Л. Босова подчеркивает, что планомерное развитие алгоритмического мышления современного школьника в рамках цифровой образовательной среды совершенно невозможно без строгого этапа формализации и графического структурирования задач [6]. В этом контексте В. В. Любимова проводит подробный сравнительный анализ существующих виртуальных симуляторов и сред для изучения робототехники, акцентируя внимание педагогов на их функциональных возможностях, дидактическом потенциале и технических ограничениях при имитации физических процессов [7]. В свою очередь В. В. Тарапата всесторонне рассматривает специфику сложного когнитивного перехода обучающихся от простых визуальных образов к многоуровневому

текстовому коду, предлагая использовать адаптированные проектные методики и стандартные блок-схемы в качестве связующего звена [8]. Вопросы успешного формирования базовых алгоритмических навыков у детей в начальной школе с применением интуитивно понятных блочно-ориентированных сред программирования, подобных Scratch, подробно описывает в своих трудах Н. А. Воробьева [9]. Исключительную важность качественного методического сопровождения педагогов и создания комплексной системы повышения квалификации при практической реализации школьных программ образовательной робототехники убедительно доказывает Н. В. Абрамовских [10]. Органично дополняет эту фундаментальную мысль Е. А. Романова, глубоко исследуя актуальные проблемы профессиональной подготовки педагогических кадров в стремительно меняющихся условиях глобальной информатизации современного образования [11]. Специфику организации дистанционного и гибридного онлайн-обучения школьников основам программирования, а также методы удержания познавательного интереса вне школьной лаборатории рассматривает в своих научных трудах Н. Ю. Куликова [12]. Специфические особенности организации инновационного научно-технического творчества учащихся в гибкой системе дополнительного образования, где робототехника выступает главным мотивирующим фактором, детально анализирует М. Д. Воронина [13]. Для непосредственной практической методической поддержки учителей на уроках информатики сегодня разработаны специализированные учебные пособия, среди которых особой практико-ориентированностью выделяется работа З. В. Яковлевой [14]. Различные технические аспекты применения облачных виртуальных симуляторов, таких как Tinkercad, при первоначальном изучении схмотехники и микроконтроллеров архитектуры Arduino на конкретных учебных примерах детально рассматривает С. Д. Коткин [15]. Фундаментальные технологические основы преподавания робототехники в основной школе, включающие поэтапное освоение механики, электроники и базовых структур программирования, последовательно закладываются в известных методических работах Д. Г. Копосова [16]. Однако глубокий анализ представленных отечественных источников показывает, что они не предлагают педагогам унифицированную платформонезависимую декомпозицию команд работы со сложными дискретными и аналоговыми датчиками.

Зарубежные исследования также подтверждают чрезвычайную актуальность поиска новых подходов к обучению программированию киберфизических систем. Проведя глубокий анализ существующих публикаций, С. Анвар констатирует существенный недостаток современных школьных курсов по робототехнике, который заключается в полном отсутствии этапа абстрактного проектирования алгоритмов [17]. В последующих эмпирических изысканиях группа исследователей под руководством С. Анвара доказывает, что повышение уровня довузовского инженерного образования требует создания совершенно иных педагогических средств [18]. В свою очередь, изучая программное обеспечение для учебных конструкторов, А. Меркурис обращает внимание на серьезный дидактический барьер, возникающий при переводе обучающихся с графических блоков на классические текстовые языки [19]. В другой своей работе А. Меркурис на основе собранных экспериментальных данных детально исследует прямое позитивное влияние блочного кодирования физических роботов на развитие базовых навыков вычислительного мышления у школьников [19]. Ученый Ф. Бенитти всесторонне изучает раскрытый образовательный потенциал робототехники в обще-

образовательных школах, параллельно фиксируя серьезные методические пробелы педагогов именно в процессе безопасного перехода учеников к сложным текстовым алгоритмам [20]. Абсолютная важность регулярного использования стандартизированных блок-схем в качестве ключевого когнитивного этапа для эффективного решения проблем и визуализации сложных логических процессов научно доказана в новейшей работе А. Гритлахаре [21]. Детальное качественное и количественное сравнение педагогических подходов к визуальному и классическому текстовому программированию в классах информатики старшей школы проводит на основе лонгитюдных наблюдений Д. Вайнтроп [22]. Турецкий исследователь М. Гелиболу статистически подтверждает наличие серьезных поведенческих и когнитивных трудностей у учащихся при неизбежном переходе от перетаскивания ярких блоков к строгому синтаксису высокоуровневых языков программирования [23]. Существенное развивающее влияние различных компьютерных визуальных сред и планшетных приложений на раннее формирование математических и логических навыков у детей младшего возраста наглядно демонстрирует в своем исследовании С. Пападакис [24]. В свою очередь, Х. Гюнеш экспериментально исследует и доказывает тот факт, что именно физическое, а не виртуальное программирование реальных мобильных роботов наиболее эффективно развивает вычислительное мышление и компетенции XXI века у учащихся средней школы [25]. Специфическую проблему рассматривает Г. Ванг, напрямую указывая на неизбежные технические и методические сложности, возникающие у преподавателей при переводе студентов с «закрытых» модульных конструкторов LEGO на «открытую» аппаратную платформу Arduino [26]. Огромный скрытый потенциал использования современных виртуальных симуляторов для поддержки преподавания алгоритмики и геометрии убедительно раскрывается коллективом авторов масштабного исследования специализированной образовательной платформы VEXcode VR [27]. Практическую специфику аккуратного внедрения инновационных образовательных подходов к преподаванию основ робототехники в строгой среде традиционной начальной школы подробно рассматривает Д. Скарадоцци [28]. Известный специалист в области раннего развития М. Берс детально описывает гармоничную интеграцию программирования в обучение дошкольников и младших школьников через игровую, практическую и исследовательскую деятельность [29]. Наконец, сложные вопросы критериальной оценки будущих технологических инноваций и эффективности применяемых методов преподавания в сфере образовательной робототехники грамотно систематизирует в своем свежем обзоре Н. Тамрин [30].

Детальный анализ представленной обширной литературы обнаруживает явное научное противоречие: большинство исследователей единогласно заявляют о необходимости промежуточного моделирования алгоритмов, однако подробная методика пошаговой декомпозиции параметров, инвариантная к используемой аппаратной платформе, в доступных источниках полностью отсутствует.

Методологическая база исследования / Methodological base of the research

На первом этапе проводилось теоретическое исследование на базе анализа технической документации и API трех наиболее распространенных в школьном образовании платформ: Arduino (язык C++), VEX (C++) и LEGO Mindstorms EV3 (MicroPython). Для достижения поставленной цели применялись следующие методы:

1. Сравнительный анализ кода: сопоставление синтаксических конструкций, отвечающих за инициализацию и опрос датчиков.

2. Метод декомпозиции: разбиение сложных программных объектов (классов и функций) на элементарные составляющие.

3. Моделирование: создание графических примитивов (блоков), отображающих логическую структуру команды.

В роли исследуемых объектов выступили три базовые категории сенсорных устройств, отличающиеся друг от друга спецификой генерации и передачи информации:

1. Контактный датчик (Touch Sensor). Рассматривается как классический генератор бинарных сигналов (фиксирует два состояния: замкнут/разомкнут).

2. Дальномер, или ультразвуковой датчик (Distance/Ultrasonic Sensor). Служит иллюстрацией устройства, выдающего числовые показатели в формате с плавающей запятой.

3. Цветовой сенсор (Color Sensor). Является образцом мультимодального оборудования. Для его корректного функционирования необходима первоначальная конфигурация (выбор режима считывания RGB-показателей, распознавания конкретного оттенка или замера интенсивности отраженного луча).

Результаты исследования / Research results

Фундаментальной составляющей любой учебной робототехнической базы выступает механизм мониторинга внешней среды, который реализуется путем регулярного считывания показаний с сенсоров. Потребность в оптимизации процесса обработки поступающей информации диктует необходимость создания понятных и эффективных инструментов. Они должны обеспечивать плавную трансформацию абстрактных алгоритмов в готовый программный код за счет точной спецификации характеристик измерительных устройств.

Для углубления исследования требуется проанализировать специфику выбранных сред разработки. Это достигается путем детального разбора доступного инструментария (функций и методов), предназначенного для управления тремя упомянутыми категориями датчиков. В качестве базы для сравнения были отобраны наиболее востребованные программные платформы: LEGO Mindstorms, VEXcode и Arduino IDE.

1. *Arduino IDE*. Взаимодействие с периферией в этой среде обязывает разработчика четко понимать физическую суть обрабатываемого сигнала (является ли он аналоговым или цифровым). Более того, зачастую возникает необходимость самостоятельного применения математических формул для конвертации «сырых» значений.

Таблица 1

Пример кода для датчиков в среде Arduino IDE

```
// 1. Датчик касания (кнопка) на цифровом пине 2
const int touchPin = 2;
// 2. Датчик расстояния (HC-SR04): пины Trig (выход) и Echo (вход)
const int trigPin = 9;
const int echoPin = 10;

void setup() {
  Serial.begin(9600); // Запуск порта для вывода данных
  pinMode(touchPin, INPUT); // Настройка пина кнопки на вход
  pinMode(trigPin, OUTPUT); // Настройка пина Trig
  pinMode(echoPin, INPUT); // Настройка пина Echo
}

void loop() {
```

```

// --- Опрос датчика касания ---
// Считываем состояние: 1 (HIGH) или 0 (LOW)
int touchState = digitalRead(touchPin);

// --- Опрос датчика расстояния ---
// Генерируем импульс 10 мкс
digitalWrite(trigPin, LOW); delayMicroseconds(2);
digitalWrite(trigPin, HIGH); delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Замеряем длину импульса (время возврата звука)
long duration = pulseIn(echoPin, HIGH);
// Пересчитываем время в сантиметры (физическая формула)
int distanceCm = duration / 58;

// Вывод данных
Serial.print("Touch: "); Serial.println(touchState);
Serial.print("Dist: "); Serial.println(distanceCm);
}

```

Примечание:

- pinMode(..., INPUT) – настройка пина на считывание данных.
- digitalRead(touchPin) – установка цифрового пина на чтение данных.
- duration / 58 – коэффициент пересчета времени.

2. *VEXcode V5 (C++)*. Среда *VEXcode* использует объектно-ориентированный подход с высоким уровнем абстракции.

Таблица 2

Пример кода для датчиков в среде *VEXcode V5*

```

#include "vex.h"
using namespace vex;

// Объявление объектов датчиков
bumper BumperA = bumper(Brain.ThreeWirePort.A); // Датчик касания
distance Distance1 = distance(PORT1); // Датчик расстояния
optical Optical1 = optical(PORT2); // Датчик цвета

int main() {
while(true) {
// --- Опрос датчика касания ---
// Метод .pressing() возвращает true (истина) или false (ложь)
bool isPressed = BumperA.pressing();

// --- Опрос датчика расстояния ---
// Метод требует указания единиц измерения (mm или inches)
double distVal = Distance1.objectDistance(mm);

// --- Опрос датчика цвета ---
// Датчик мультимодальный: можно узнать имя цвета или яркость
if (Optical1.color() == red) {
// Действие, если увидели красный
}
// Или считываем яркость (0-100%)
double brightness = Optical1.brightness();

task::sleep(20);
}
}

```

```
}
}
}
```

Примечание:

- BumperA.pressing() - получение состояния нажатия кнопки.
- Distance1.objectDistance(mm) - получение значения расстояния с ключевой особенностью: необходимо явно указать единицы измерения (аргумент функции).
- Optical1.color() и Optical1.brightness() - получение значений цвета и яркости; один датчик имеет разные методы для разных режимов работы.

3. LEGO Mindstorms EV3 (MicroPython). Работа с методами без низкоуровневого обращения к портам.

Таблица 3

Пример кода для датчиков в среде LEGO Mindstorms EV3

```
from pybricks.hubs import EV3Brick
from pybricks.ev3devices import TouchSensor, UltrasonicSensor, ColorSensor
from pybricks.parameters import Port, Color

# Инициализация объектов (связь софта с портами)
ts = TouchSensor(Port.S1) # Датчик касания
us = UltrasonicSensor(Port.S2) # Ультразвуковой датчик
cs = ColorSensor(Port.S3) # Датчик цвета

while True:
    # --- Опрос датчика касания ---
    # Возвращает True/False
    if ts.pressed():
        print("Нажато!")

    # --- Опрос датчика расстояния ---
    # По умолчанию возвращает миллиметры (int)
    dist_mm = us.distance()

    # --- Опрос датчика цвета ---
    # Режим определения цвета (возвращает объект Color.RED и т.д.)
    detected_color = cs.color()
    # Режим отраженного света (яркость 0-100%)
    reflection = cs.reflection()
```

Примечание:

- TouchSensor(Port.S1) - создание объекта и инициализация порта.
- us.distance() - метод по умолчанию работает в миллиметрах, переключение на дюймы или сантиметры требует математических операций.
- cs.color() и cs.reflection() - методы разделяют режимы работы датчика (распознавание цвета или движение по линии).

Результаты сравнения:

1. В качестве базового параметра у всех датчиков выступает порт подключения устройства.
2. Для датчиков в различных средах нет разницы для входных данных, то есть программирование устройств является типовым и поддается шаблонизации.

Универсальная модель декомпозиции

Проведенный сравнительный анализ позволяет сделать ключевой вывод: несмотря на кардинальные различия в синтаксисе (процедурный C++, объектно-ориентированный C++, скриптовый Python) и уровнях аппаратной абстракции, логика инициализации и настройки параметров датчиков во всех средах концептуально идентична. Процесс программного опроса любого сенсора можно разложить на три обязательных параметра, которые должны быть определены обучающимся до написания кода:

- Порт (аппаратный интерфейс): определение точки физического подключения периферийного устройства к микроконтроллеру (например, Pin 2, PORT1, Port.S1).

- Режим (функциональное назначение): определение того, какую именно физическую величину или состояние в данный момент фиксирует датчик. Для простых датчиков этот параметр статичен (чтение состояния), для сложных (мультимодальных) – динамичен (определение цвета, измерение интенсивности света, измерение RGB-составляющих).

- Тип данных (программный интерфейс): понимание того, в каком формате данные поступят в программу для дальнейшей обработки (бинарное логическое значение, целое число, число с плавающей точкой, специфический объект класса).

На основе выявленных инвариантных параметров нами предложено отказаться от использования абстрактного параллелограмма «Получить данные» из классических стандартов составления блок-схем. Вместо него разработана система декомпозированных графических блоков (см. рисунок).

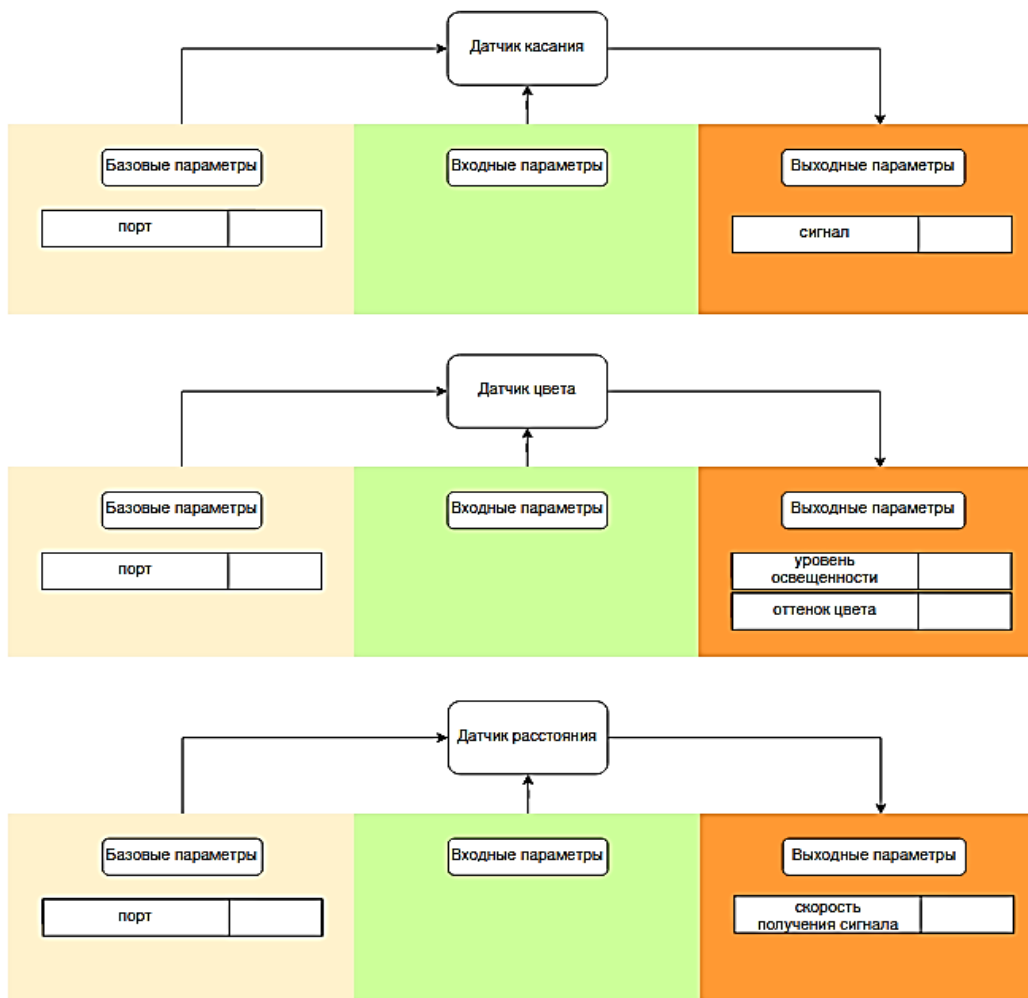
Разработанный графический примитив имеет составную архитектуру. В его верхней части прописывается категория сенсора, а внутреннее пространство разделено на строго регламентированные секции: «Аппаратный порт», «Функциональный режим» и «Тип переменной». Продемонстрируем практическое использование данной концепции:

- Датчик касания: в схеме четко прописывается интерфейс подключения (допустим, D2), в качестве неизменного режима устанавливается «нажатие», а формат данных задается как булевый (1/0 или True/False). Благодаря этому обучающийся интуитивно понимает, что полученное значение логичнее всего встраивать в конструкцию условного перехода (оператор if).

- Дальномер (датчик расстояния): визуальный элемент включает информацию о контактах (например, пара Trig/Echo для плат Arduino), настройку «измерение дистанции» и числовой формат переменной (вещественный или целочисленный). Обязательным условием является указание метрической системы (дюймы, миллиметры или сантиметры).

- Мультимодальный оптический датчик (цвета): именно здесь методика раскрывается в полной мере. В зависимости от поставленной алгоритмической цели один реальный физический компонент визуализируется в схеме совершенно разными конфигурациями.

Если робот движется по черной линии, в блоке задается режим «Отраженный свет», а тип данных указывается как числовой процент (0–100%). Если робот сортирует кубики, задается режим «Распознавание цвета», а тип данных определяется как специализированный идентификатор (Color.RED, Color.BLUE).



Декомпозиция команд управления датчиками

Обоснование методической эффективности декомпозиции команд управления датчиками

Теоретический анализ функциональных возможностей сред программирования позволяет выдвинуть гипотезу о высокой методической ценности предлагаемого подхода для школьного курса информатики и технологии. Методика позиционируется как промежуточное звено (scaffolding), обеспечивающее переход от наглядно-образного представления алгоритмов к абстрактно-логическому.

Во-первых, предлагаемый метод направлен на решение проблемы «черного ящика». При использовании стандартных графических блоков ученик часто оперирует абстракцией «Получить данные», не анализируя физическую природу сигнала. Декомпозиция команды на элементарные составляющие (порт, режим работы, тип данных) создает условия для формирования у обучающихся инженерного мышления и понимания аппаратно-программных связей.

Во-вторых, визуализация параметров в виде отдельных блоков на этапе проектирования может служить средством профилактики синтаксических ошибок. Предполагается, что, когда школьник переходит к написанию кода на C++ или Python, наличие детализированной схемы снизит когнитивную нагрузку, так как логическая структура команды уже сформирована. В-третьих, унификация подхода обеспечивает метапредметность. Независимо от платформы (Arduino, VEX, LEGO), принцип декомпозиции остается неизменным, что позволяет педагогу фокусироваться на фундаментальных принципах работы сенсорных систем, а не на особенностях конкретного конструктора.

Заключение / Conclusion

В ходе исследования была достигнута поставленная цель: разработан универсальный метод декомпозиции команд управления датчиками (касания, расстояния, цвета), инвариантный к образовательным робототехническим платформам. На основании сравнительного анализа сред разработки (Arduino IDE, VEXcode, LEGO Mindstorms EV3) были решены следующие задачи:

- Выявлены общие паттерны инициализации и опроса сенсоров, сводящиеся к определению порта, режима работы и формата возвращаемых данных.
- Разработаны унифицированные графические блоки, позволяющие визуализировать логику работы датчиков до этапа написания программного кода.
- Теоретически обоснована необходимость внедрения этапа детализированного блок-схемного моделирования для минимизации барьера при переходе от визуальных сред к текстовым языкам высокого уровня. Дальнейшим направлением исследования станет экспериментальная проверка эффективности методики в школьных условиях и автоматизация процесса генерации программного кода на основе разработанных схем.

Ссылки на источники / References

1. Параскевов А. В. Современная робототехника в России: реалии и перспективы // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. – 2014. – № 104 (10). – URL: <http://ej.kubagro.ru/2014/10/pdf/116.pdf>
2. Никитина Т. В. Образовательная робототехника как направление инженерно-технического творчества школьников: учеб. пособие. – Челябинск: Изд-во Челяб. гос. пед. ун-та, 2014. – 169 с.
3. Хеннер Е. К. Вычислительное мышление // Образование и наука. – 2016. – № 2. – С. 18–33. DOI: 10.17853/1994-5639-2016-2-18-33.
4. Lye S. Y., Koh J. H. L. Review on teaching and learning of computational thinking through programming: What is next for K-12? // Computers in Human Behavior. – 2014. – Vol. 41. – P. 51–61. DOI: 10.1016/j.chb.2014.09.012.
5. Wing J. M. Computational thinking // Communications of the ACM. – 2006. – Vol. 49, Iss. 3. – P. 33–35. DOI: 10.1145/1118178.1118215.
6. Босова Л. Л. Цифровые навыки современного школьника и возможности их формирования в школьном курсе информатики // Информатика в школе. – 2020. – № 1 (7). – С. 5–9. DOI: 10.32517/2221-1993-2020-19-7-5-9.
7. Любимова В. В. Виртуальная робототехника в реальности: большая подборка программ // Занимательная робототехника: [сайт]. – 2020. – URL: <https://edurobots.org/2020/05/virtual-toolkits/>
8. Тарапата В. В., Самылкина Н. Н. Робототехника в школе: методика, программы, проекты. – М.: Лаборатория знаний, 2017. – 109 с.
9. Воробьева Н. А. Развитие алгоритмического мышления у учащихся начальной школы с использованием системно-ориентированной среды программирования Scratch // Вестник ПГГПУ. Серия № 1. Психологические и педагогические науки. – 2023. – С. 59–65.
10. Абрамовских Н. В., Асланова Е. С. Методическое сопровождение педагогов начальной школы по реализации образовательных программ с применением робототехники // Вестник СГПУ. – 2021. – № 4 (73). – С. 61–69.
11. Романова Е. А., Кузнецов В. А., Тореева Т. А. Подготовка педагогических кадров в условиях информатизации образования // Вестник Московского университета. Серия 20. Педагогическое образование. – 2018. – № 2. – С. 78–84.
12. Куликова Н. Ю., Шемелова Т. В., Цымбалюк Г. В. Анализ опыта онлайн-обучения основам программирования и робототехнике школьников // Грани познания. – 2021. – № 6 (77). – С. 174–180.
13. Воронина М. Д. Особенности организации инновационного научно-технического творчества учащихся в системе дополнительного образования // Приоритетные направления развития науки и образования: сб. материалов междунар. науч.-практ. конф. – Чебоксары: Изд. дом «Среда», 2024. – С. 2–4.
14. Яковлева З. В. Образовательная робототехника на уроках информатики и ИКТ. 5 класс: учеб.-метод. пособие для слушателей курса. – М.: Издательство «Перо», 2014. – 48 с.
15. Коткин С. Д. Особенности использования Tinkercad в обучении программированию микроконтроллеров Arduino // Материалы XVII Международной научно-практической конференции «Технологическое обучение школьников и профессиональное образование в России и за рубежом». – 2023. – № 1 (82).
16. Копосов Д. Г. Технология. Робототехника. 5–9 классы: учеб. пособие. – М.: БИНОМ. Лаборатория знаний, 2020. – 96 с.

17. Anwar S., Bascou N. A., Menekse M., Kardgar A. A Systematic Review of Studies on Educational Robotics // Journal of Pre-College Engineering Education Research (J-PEER). – 2019. – Vol. 9, Iss. 2. – Article 2. DOI: 10.7771/2157-9288.1223.
 18. Merkouris A., Chorianopoulos K. Programming environments for educational robotics: A systematic review // Robotics. – 2022. – Vol. 11, Iss. 6. – P. 120. DOI: 10.3390/robotics11060120.
 19. Merkouris A., Chorianopoulos K. Coding with robots vs. block-based coding: effects on K-12 students' computational thinking skills // Proceedings of the 19th International Conference on Informatics (PCI '19). – New York: ACM, 2019. – Article 14. – P. 1–6. DOI: 10.1145/3364335.3364369.
 20. Benitti F. B. V. Exploring the educational potential of robotics in schools: A systematic review // Computers & Education. – 2012. – Vol. 58, Iss. 3. – P. 978–988. DOI: 10.1016/j.compedu.2011.10.006.
 21. Ghritlahare A. The Role of Flowcharts in Problem Solving and Process Visualization [Preprint] // ResearchGate. – 2023. DOI: 10.5281/zenodo.15358152
 22. Weintrop D., Wilensky U. Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms // Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. – New York: ACM, 2017. – P. 193–198. DOI: 10.1145/3017680.3017707.
 23. Gelibolu M. F., Cavus N. Block-based versus text-based programming: a comparison of learners' programming behaviors, computational thinking skills, and attitudes toward programming // Education and Information Technologies. – 2024. DOI: 10.1007/s10639-023-12423-3.
 24. Papadakis S., Kalogiannakis M., Zaranis N. The effectiveness of computer and tablet assisted intervention in early childhood students' understanding of numbers // International Journal of Child-Computer Interaction. – 2018. – Vol. 16. – P. 11–19. DOI: 10.1016/j.ijcci.2017.12.001.
 25. Güneş H., Küçük S. The impact of visual and physical programming-based science activities on middle school students' 21st century and computational thinking skills // Bartın University Journal of Faculty of Education. – 2025. – Vol. 14, № 3. – P. 858–882. DOI: 10.14686/buefad.1670825.
 26. Wang G., Chen J. From LEGO to Arduino: Enhancement of ECE Freshman Design with Practical Applications // 2016 ASEE Annual Conference & Exposition Proceedings. – New Orleans: ASEE, 2016. DOI: 10.18260/p.26966.
 27. Cardona-Reyes H., Federico Parra-Gonzalez E., Murillo Alfaro R. VEXCODE VR: A Virtual Tool as Support in the Teaching of Analytical Geometry // CLEI electronic journal. – May 2025. – 28(2). DOI: <https://doi.org/10.19153/10.19153/cleiej.28.2>
 28. Scaradozzi D., Sorbi L., Pedale A. et al. Teaching Robotics at the Primary School: An Innovative Approach // Procedia – Social and Behavioral Sciences. – 2015. – Vol. 174. – P. 3838–3846. DOI: 10.1016/j.sbspro.2015.01.1122.
 29. Bers M. U., Flannery L., Kazakoff E. R., Sullivan A. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum // Computers & Education. – 2014. – Vol. 72. – P. 145–157. DOI: 10.1016/j.compedu.2013.10.020.
 30. Thamrin N. M. Future Innovations and Evaluation Methods in Robotics Education // Educational Robotics. EDUBOTS 2024. – Singapore: Springer, 2025. – P. 73–89. DOI: 10.1007/978-981-95-3548-4_6.
-
1. Paraskevov, A. V. (2014). "Sovremennaya robototekhnika v Rossii: realii i perspektivy" [Modern robotics in Russia: realities and prospects], *Politematicheskij setevoy elektronnyj nauchnyj zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta*, № 104 (10). Available at: <http://ej.kubagro.ru/2014/10/pdf/116.pdf> (in Russian).
 2. Nikitina, T. V. (2014). *Obrazovatel'naya robototekhnika kak napravlenie inzhenerno-tehnicheskogo tvorchestva shkol'nikov* [Educational robotics as a direction of engineering and technical creativity of schoolchildren]: *ucheb. posobie*, Izd-vo Chelyab. gos. ped. un-ta, Chelyabinsk, 169 p. (in Russian).
 3. Henner, E. K. (2016). "Vychislitel'noe myshlenie" [Computational thinking], *Obrazovanie i nauka*, № 2, pp. 18–33. DOI: 10.17853/1994-5639-2016-2-18-33 (in Russian).
 4. Lye, S. Y., & Koh, J. H. L. (2014). "Review on teaching and learning of computational thinking through programming: What is next for K-12?", *Computers in Human Behavior*, vol. 41, pp. 51–61. DOI: 10.1016/j.chb.2014.09.012 (in English).
 5. Wing, J. M. (2006). "Computational thinking", *Communications of the ACM*, vol. 49, iss. 3, pp. 33–35. DOI: 10.1145/1118178.1118215 (in English).
 6. Bosova, L. L. (2020). "Cifrovye navyki sovremennogo shkol'nika i vozmozhnosti ih formirovaniya v shkol'nom kurse informatiki" [Digital skills of a modern schoolchild and the opportunities for their development in a school computer science course], *Informatika v shkole*, № 1 (7), pp. 5–9. DOI: 10.32517/2221-1993-2020-19-7-5-9 (in Russian).
 7. Lyubimova, V. V. (2020). "Virtual'naya robototekhnika v real'nosti: bol'shaya podborka program" [Virtual robotics in reality: a large collection of programs], *Zanimatel'naya robototekhnika: [sajt]*. Available at: <https://edurobots.org/2020/05/virtual-toolkits/> (in Russian).
 8. Tarapata, V. V., & Samylkina, N. N. (2017). *Robototekhnika v shkole: metodika, programmy, proekty* [Robotics in School: Methods, Programs, and Projects], Laboratoriya znaniy, Moscow, 109 p. (in Russian).
 9. Vorob'eva, N. A. (2023). "Razvitie algoritmicheskogo myshleniya u uchashchihsya nachal'noj shkoly s ispol'zovaniem sistemno-orientirovannoj sredy programmirovaniya Scratch" [Developing algorithmic thinking in primary school students using the system-oriented programming environment Scratch], *Vestnik PGGPU. Seriya № 1. Psichologicheskie i pedagogicheskie nauki*, pp. 59–65 (in Russian).

10. Abramovskih, N. V., & Aslanova, E. S. (2021). "Metodicheskoe soprovozhdenie pedagogov nachal'noj shkoly po realizacii obrazovatel'nyh programm s primeneniem robototekhniki" [Methodological support for primary school teachers in the implementation of educational programs using robotics], *Vestnik SGPU*, № 4 (73), pp. 61–69 (in Russian).
11. Romanova, E. A., Kuznecov, V. A., & Toreeva, T. A. (2018). "Podgotovka pedagogicheskikh kadrov v usloviyah informatizacii obrazovaniya" [Training of teaching staff in the context of informatization of education], *Vestnik Moskovskogo universiteta. Seriya 20. Pedagogicheskoe obrazovanie*, № 2, pp. 78–84 (in Russian).
12. Kulikova, N. Yu., Shemelova, T. V., & Cymbalyuk, G. V. (2021). "Analiz opyta onlajn-obucheniya osnovam programmirovaniya i robototekhnike shkol'nikov" [Analysis of the experience of online teaching of programming basics and robotics to schoolchildren], *Grani poznaniya*, № 6 (77), pp. 174–180 (in Russian).
13. Voronina, M. D. (2024). "Osobennosti organizacii innovacionnogo nauchno-tekhnicheskogo tvorchestva uchashchihsya v sisteme dopolnitel'nogo obrazovaniya" [Characteristics of the organization of innovative scientific and technical creativity of students in the system of additional education], *Prioritetnye napravleniya razvitiya nauki i obrazovaniya: sbornik materialov mezhdunarodnoj nauchno-prakticheskoy konferencii*, Izd. dom "Sreda", Cheboksary, pp. 2–4 (in Russian).
14. Yakovleva, Z. V. (2014). *Obrazovatel'naya robototekhnika na urokah informatiki i IKT. 5 klass [Educational robotics in computer science and ICT classes. 5th grade]: ucheb.-metod. posobie dlya slushatelej kursa*, Izdatel'stvo "Pero", Moscow, 48 p. (in Russian).
15. Kotkin, S. D. (2023). "Osobennosti ispol'zovaniya Tinkercad v obuchenii programmirovaniyu mikrokontrollerov Arduino" [Using Tinkercad to Teach Arduino Microcontroller Programming], *Materialy XVII Mezhdunarodnoj nauchno-prakticheskoy konferencii "Tekhnologicheskoe obuchenie shkol'nikov i professional'noe obrazovanie v Rossii i za rubezhom"*, № 1 (82) (in Russian).
16. Kuposov, D. G. (2020). *Tekhnologiya. Robototekhnika. 5–9 klassy [Technology. Robotics. Grades 5–9]: ucheb. posobie*, BINOM. Laboratoriya znaniy, Moscow, 96 p. (in Russian).
17. Anwar, S., Bascou, N. A., Menekse, M., & Kardgar, A. (2019). "A Systematic Review of Studies on Educational Robotics", *Journal of Pre-College Engineering Education Research (J-PEER)*, vol. 9, iss. 2, Article 2. DOI: 10.7771/2157-9288.1223 (in English).
18. Merkouris, A., & Chorianopoulos, K. (2022). "Programming environments for educational robotics: A systematic review", *Robotics*, vol. 11, iss. 6, p. 120. DOI: 10.3390/robotics11060120 (in English).
19. Merkouris, A., & Chorianopoulos, K. (2019). "Coding with robots vs. block-based coding: effects on K-12 students' computational thinking skills", *Proceedings of the 19th International Conference on Informatics (PCI '19)*, ACM, New York, Article 14, pp. 1–6. DOI: 10.1145/3364335.3364369 (in English).
20. Benitti, F. B. V. (2012). "Exploring the educational potential of robotics in schools: A systematic review", *Computers & Education*, vol. 58, iss. 3, pp. 978–988. DOI: 10.1016/j.compedu.2011.10.006 (in English).
21. Ghriltahare, A. (2023). "The Role of Flowcharts in Problem Solving and Process Visualization [Preprint]", *ResearchGate*. DOI: 10.5281/zenodo.15358152 (in English).
22. Weintrop, D., & Wilensky, U. (2017). "Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms", *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, New York: ACM, pp. 193–198. DOI: 10.1145/3017680.3017707 (in English).
23. Gelibolu, M. F., & Cavus, N. (2024). "Block-based versus text-based programming: a comparison of learners' programming behaviors, computational thinking skills, and attitudes toward programming", *Education and Information Technologies*. DOI: 10.1007/s10639-023-12423-3 (in English).
24. Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2018). "The effectiveness of computer and tablet assisted intervention in early childhood students' understanding of numbers", *International Journal of Child-Computer Interaction*, vol. 16, pp. 11–19. DOI: 10.1016/j.ijcci.2017.12.001 (in English).
25. Güneş, H., & Küçük, S. (2025). "The impact of visual and physical programming-based science activities on middle school students' 21st century and computational thinking skills", *Bartın University Journal of Faculty of Education*, vol. 14, № 3, pp. 858–882. DOI: 10.14686/buefad.1670825 (in English).
26. Wang, G., & Chen, J. (2016). "From LEGO to Arduino: Enhancement of ECE Freshman Design with Practical Applications", *2016 ASEE Annual Conference & Exposition Proceedings*, ASEE, New Orleans. DOI: 10.18260/p.26966 (in English).
27. Cardona-Reyes, H., Federico Parra-Gonzalez, E., & Murillo Alfaro, R. (2025). "VEXCODE VR: A Virtual Tool as Support in the Teaching of Analytical Geometry", *CLEI electronic journal*, May, 28(2). DOI: <https://doi.org/10.19153/10.19153/cleiej.28.2> (in English).
28. Scaradozzi, D., Sorbi, L., Pedale, A. et al. (2015). "Teaching Robotics at the Primary School: An Innovative Approach", *Procedia – Social and Behavioral Sciences*, vol. 174, pp. 3838–3846. DOI: 10.1016/j.sbspro.2015.01.1122 (in English).
29. Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). "Computational thinking and tinkering: Exploration of an early childhood robotics curriculum", *Computers & Education*, vol. 72, pp. 145–157. DOI: 10.1016/j.compedu.2013.10.020 (in English).
30. Thamrin, N. M. (2025). "Future Innovations and Evaluation Methods in Robotics Education", *EduBOTS 2024*, Springer, Singapore, pp. 73–89. DOI: 10.1007/978-981-95-3548-4_6 (in English).