

Программирование на C++ с роботом РУБИ

Программа курса: 36 очных встреч по 120 минут: 45 минут занятий + 30 минут перерыва + 45 минут занятий.

Возраст: 13–17 лет. Уровень начальный; опыт текстового программирования полезен, но не обязателен. Для младших участников с опытом Python темп определяется входной пробой.

Учебный объём: 72 академических часа по 45 минут, или 54 астрономических часа. Перерывы — 18 астрономических часов; общее время присутствия — 72 астрономических часа.

Группа: 6–10 человек; каждому нужен компьютер. На занятиях с устройством — один проверенный РУБИ на пару.

Ритм: одна встреча в неделю — 36 учебных недель без учёта каникул; две встречи — 18 учебных недель.

Итог: личный исходный код на C++, журнал испытаний и работающая миссия с физическим РУБИ.

1. Идея и границы курса

Подросток учится писать небольшие программы на C++: получает данные, вычисляет, выбирает действие, повторяет команды, оформляет функции и проверяет ошибки. Сначала код работает на компьютере с текстовыми данными и экранной моделью робота. Затем знакомые решения переносятся в скетч для конкретного контроллера РУБИ: световой и звуковой сигнал, показания датчика, ограниченное движение колёс на подставке.

Две среды исполнения. На компьютере используется выбранный педагогом компилятор C++ и консоль. На показанном в исходных материалах контроллере Arduino Nano выполняется скомпилированный скетч C++ с библиотеками Arduino. Это не один и тот же исполняемый файл: алгоритм переносится с учётом ограничений памяти, времени и доступных библиотек. `std::cout`, `std::string` и `std::vector` относятся к компьютерному этапу; в обязательных заданиях для контроллера применяются простые типы, функции, массивы фиксированного размера и проверенные функции работы с устройством.

Курс посвящён программированию. Самостоятельная сборка электроники, пайка, настройка питания, разработка универсальной библиотеки для всех ревизий РУБИ и точная навигация по полу не входят в обязательный результат. Для всех обязательных заданий нужны проверенные на фактической сборке световой сигнал, звуковой сигнал, датчик HC-SR04 и базовый привод. Bluetooth, микрофон, матрица, захват и другие надстройки не предполагаются установленными.

2. Цель, задачи и результаты

Цель: научить подростка создавать, проверять и объяснять небольшие программы на C++ и применять их для безопасного управления базовыми функциями РУБИ.

Учебные задачи: освоить структуру программы, переменные и типы, ввод и вывод, условия, циклы, функции, массивы и простую обработку ошибок; научиться компилировать и читать диагностические сообщения; перенести правило выбора из консольной модели в скетч; получить и проверить показание датчика; управлять сигналом и ограниченным движением робота.

Развивающие задачи: формулировать ожидаемый результат до запуска, проверять обычные, граничные и ошибочные данные, менять одну причину за раз, фиксировать вывод по нескольким испытаниям.

Практические задачи: соблюдать порядок работы с общим устройством, лично изменять и загружать свой код под наблюдением педагога, уверенно использовать штатную и физическую остановку.

К концу курса обучающийся:

- пишет, компилирует и запускает консольную программу с условием, циклом, функцией и массивом; объясняет назначение основных строк своего кода;
- различает ошибку компиляции, ошибку выполнения и неверный результат алгоритма; исправляет обнаруженную ошибку и проверяет правку повторным запуском;
- переносит правило «вход → проверка → решение → действие» в скетч для установленного контроллера;

- получает показание датчика, различает допустимое измерение и сбой или тайм-аут, выбирает безопасную реакцию;
- включает световой или звуковой ответ, а на подставке выполняет короткую моторную команду с ограничением времени и остановкой;
- сохраняет исходный код и журнал не менее пяти испытаний итогового проекта, объясняет входные данные, правило, действие и ограничение решения.

Скорость набора и запоминание синтаксиса не оцениваются. Обучающийся может пользоваться памяткой команд и стартовым шаблоном соединения с РУБИ; собственными остаются правило, функция, проверка и объяснение.

3. Подготовка площадки и технический допуск

До начала потока педагог выбирает и фиксирует версию компилятора C++ для компьютеров, среду сборки и загрузки скетчей, модель контроллера, его ревизию и паспорт каждого РУБИ: схема питания, назначение выводов, установленный датчик, свет, звук, драйвер моторов, штатная и физическая остановка. Для каждого экземпляра готовятся проверенные стартовые скетчи `led`, `beep`, `distance`, `stop` и ограниченное движение. Имена учебных функций в карточках являются локальным контрактом курса, а не утверждением о заводском API.

Педагог проверяет на каждом комплекте: сборку и загрузку скетча; свет и звук; измерение известного расстояния и ответ при отсутствии эха; безопасное состояние после включения и перезагрузки; остановку при истечении времени движения, ошибке датчика и потере управляющего сигнала. Движение допускается только после проверки схемы питания, подставки и физического выключателя. Если одновременное питание через USB и аккумулятор не подтверждено схемой конкретной сборки, его не применяют.

Условие моторных занятий: команда привода имеет верхний предел длительности, а остановка реализована в скетче и проверена независимо от продолжения основного сценария. Прерывание программы на компьютере или потеря связи не должны оставлять колёса во вращении. Если это не подтверждено на конкретном экземпляре, занятия 27–30 выполняются на экранной модели; личная практика с приводом переносится после устранения неисправности и не засчитывается демонстрацией педагога.

Педагог заранее подготавливает экранную модель с записанными показаниями датчика. Она позволяет заниматься без домашнего робота и продолжать программирование при временном отказе оборудования. Для задач с реальным РУБИ у каждого ученика остаётся личный файл. Партнёры меняются ролями между двумя учебными блоками; каждый лично вносит изменение, запускает программу и объясняет результат.

4. Ритм одной встречи

Время	Действие
0–5 мин	Миссия и повторение знакомого приёма
5–13 мин	Короткий показ нового понятия и ожидаемого результата
13–35 мин	Личная программа: написать или изменить, собрать, запустить
35–45 мин	Проверить обычный и граничный либо ошибочный случай, сохранить код
45–75 мин	Перерыв вне рабочего места; РУБИ остановлен и обесточен по процедуре педагога
75–80 мин	Возврат к задаче; при работе в паре — смена ролей
80–110 мин	Практическая миссия с одним усложнением или реальными данными РУБИ
110–120 мин	Повторная проверка, короткое объяснение, запись следующего шага и уборка

На сложных занятиях используется минимальный рабочий шаблон с готовыми подключениями и назначениями выводов. Обязательная часть выполняется до свободного усложнения. Домашние задания добровольны и

выполняются без физического робота.

5. Учебный план: 36 встреч

В каждом занятии указан минимальный наблюдаемый результат. **Экран** — консольная программа или модель на компьютере; **РУБИ** — личная практика с физическим устройством. Каждый из шести этапов содержит 12 академических часов.

Этап 1. Первые программы и выбор — занятия 1–6

1. РУБИ знакомится с оператором. Понятие: исходный файл, компиляция, `main`, `std::cout`. Первый блок: собрать и запустить готовое приветствие, найти точку входа. Второй блок: изменить три строки сообщения и имя миссии, сохранить и повторно собрать файл. **Результат:** личная консольная программа; **экран**.

2. Энергия миссии. Понятие: переменные, `int`, присваивание, арифметика. Первый блок: рассчитать условный остаток энергии. Второй блок: предсказать результат для двух наборов чисел, найти ошибку в имени переменной. **Результат:** два верных расчёта и объяснённая правка; **экран**.

3. Расстояние и время. Понятие: `double`, единицы измерения, деление и округление вывода. Первый блок: перевести метры в сантиметры и рассчитать условное время. Второй блок: проверить нулевую и дробную дистанции; указать единицы в выводе. **Результат:** калькулятор параметров с тремя проверками; **экран**.

4. Команда пользователя. Понятие: `std::cin`, ввод и проверка успешного чтения. Первый блок: запросить дистанцию и вывести её в инструкции оператору. Второй блок: проверить число и нечисловой ввод по готовой заготовке; вывести понятное сообщение при ошибке. **Результат:** программа не использует неинициализированное значение после неверного ввода; **экран**.

5. Если предмет близко. Понятие: сравнения, `if`, `else`, логический тип. Первый блок: выбрать сигнал по введённой дистанции. Второй блок: испытать значение ниже порога, ровно на пороге и выше; изменить порог. **Результат:** две ветви и три записанных ожидания; **экран**.

6. Мастерская первого этапа. Понятие: различие ошибки компиляции и неверного результата. Первый блок: исправить две короткие программы с ошибкой скобки, типа или условия. Второй блок: самостоятельно собрать программу «расстояние → ответ» и объяснить проверку на границе. **Результат:** личный файл и карта первой проверки; **экран**.

Этап 2. Повторения и функции — занятия 7–12

7. Повторяем действия. Понятие: `for`, счётчик, границы цикла. Первый блок: вывести последовательность шагов миссии. Второй блок: проверить 0, 1 и 5 повторений, исправить лишний проход. **Результат:** цикл с предсказуемым числом шагов; **экран**.

8. Сигнал по расписанию. Понятие: вложение условия в цикл. Первый блок: создать текстовый рисунок из коротких и длинных сигналов. Второй блок: изменить число повторов и убедиться, что оно ограничено. **Результат:** две различные последовательности без бесконечного цикла; **экран**.

9. Своя команда. Понятие: объявление и вызов функции, параметр. Первый блок: вынести повторяющийся вывод в функцию. Второй блок: написать `reportDistance(double value)` и вызвать её с двумя значениями. **Результат:** собственная функция и два проверенных вызова; **экран**.

10. Результат функции. Понятие: `return`, тип результата, ранний выход. Первый блок: написать функцию классификации дистанции. Второй блок: проверить обычное, граничное и недопустимое значение. **Результат:** функция возвращает различные состояния без обращения к роботу; **экран**.

11. Набор измерений. Понятие: массив фиксированного размера, индекс, перебор. Первый блок: хранить пять учебных измерений. Второй блок: найти минимум и максимум, проверить первый и последний элемент. **Результат:** короткий отчёт по пяти значениям без выхода за границы массива; **экран**.

12. Таблица испытаний. Понятие: функция плюс массив и повторная проверка. Первый блок: применить классификатор ко всем пяти значениям. Второй блок: изменить одно измерение, сравнить ожидаемые и

фактические ответы, показать файл партнёру. **Результат:** личный отчёт и карта второй проверки; **экран.**

Этап 3. Данные, состояния и проверка — занятия 13–18

13. Команды текстового пульта. Понятие: `std::string`, сравнение строк. Первый блок: распознать «старт», «стоп», «помощь» в консоли. Второй блок: обработать неизвестное слово как ошибку, не как команду движения. **Результат:** четыре проверенных варианта ввода; **экран.**

14. Дневник миссии. Понятие: запись в текстовый CSV-файл на компьютере. Первый блок: сохранить пять значений с заголовком и единицей измерения. Второй блок: открыть файл и проверить число строк и пропущенное значение по заготовке. **Результат:** личный файл данных и его проверка; **экран.**

15. Правило без сигнала датчика. Понятие: допустимый диапазон, код состояния ошибки. Первый блок: отличить число от «нет измерения» в подготовленном наборе данных. Второй блок: функция возвращает «близко», «далеко» или «ошибка»; проверить три случая. **Результат:** безопасное правило выбора; **экран.**

16. Состояния РУБИ. Понятие: перечисление состояний и переходы. Первый блок: нарисовать схему «ожидает — предупреждает — ошибка». Второй блок: пройти по пяти записанным измерениям и вывести изменения состояния. **Результат:** текстовая модель поведения; **экран.**

17. Ограниченный опрос. Понятие: `while`, счётчик и условие выхода. Первый блок: обработать записанные показания не более заданного числа раз. Второй блок: проверить окончание данных, ошибку и явную остановку. **Результат:** цикл завершается во всех трёх ситуациях; **экран.**

18. Проверка у границы. Понятие: тестовый набор и точечная правка. Первый блок: составить таблицу ниже, равно и выше порога, добавить недопустимое значение. Второй блок: запустить тесты, исправить одну ошибку и сохранить версии «до» и «после». **Результат:** проверенный классификатор и карта третьей проверки; **экран.**

Этап 4. C++ на контроллере — занятия 19–24

19. Первый скетч. Понятие: `setup()`, `loop()`, загрузка в установленный контроллер. Первый блок: педагог показывает проверенный неподвижный РУБИ и порядок безопасной загрузки. Второй блок: каждый меняет длительность светового сигнала в стартовом скетче, загружает и наблюдает результат. **Результат:** личное изменение и запуск светового сигнала; **РУБИ**, неподвижный.

20. Световой код. Понятие: функция вывода и ограниченный повтор. Первый блок: перевести экранную последовательность занятия 8 в проверенный скетч. Второй блок: партнёры меняются, каждый задаёт собственное число вспышек в допустимых пределах. **Результат:** два различных световых сообщения; **РУБИ**, неподвижный.

21. Звуковой ответ. Понятие: параметры функции и допустимая длительность сигнала. Первый блок: испытать два коротких звука на проверенном модуле. Второй блок: написать функцию «готов» или «внимание» со светом и звуком; остановить оба сигнала. **Результат:** две узнаваемые реакции; **РУБИ**, неподвижный.

22. Дальномер. Понятие: измерение с тайм-аутом и единицей измерения. Первый блок: педагог показывает проверенный вызов HC-SR04 и отдельный код ошибки. Второй блок: каждый снимает три показания у мишени на известной дистанции, сравнивает с линейкой и отмечает сбой отдельно. **Результат:** личный журнал трёх измерений; **РУБИ**, неподвижный.

23. Предупреждение о препятствии. Понятие: перенос проверенного условия в скетч. Первый блок: на записанных значениях повторить правило занятия 15. Второй блок: подключить реальный дальномер; близкий предмет даёт свет или звук, дальний — тишину, ошибка — отдельную безопасную реакцию. **Результат:** три различных случая; **РУБИ**, неподвижный.

24. Настольный сторож. Понятие: соединение функции, состояния и ограниченного опроса. Первый блок: отдельно испытать датчик и ответы. Второй блок: собрать цикл «измерить → проверить → выбрать → ответить», ограничить длительность работы и сохранить скетч. **Результат:** работающий неподвижный сторож и карта четвёртой проверки; **РУБИ.**

Этап 5. Проверенное движение — занятия 25–30

25. Маршрут в модели. Понятие: последовательность и декомпозиция. Первый блок: выложить на карточках «вперёд — стоп — поворот — стоп». Второй блок: смоделировать маршрут в консоли и исправить порядок команд.

Результат: текстовая модель с явной остановкой; **экран**.

26. Ограничение привода. Понятие: проверка аргумента и время последней команды. Первый блок: написать экранную модель, отвергающую слишком длинное движение. Второй блок: проверить обычную команду,

превышение предела и сбой датчика; во всех завершениях ожидается остановка. **Результат:** три зарегистрированных проверки правила; **экран**.

27. Колёса на подставке. Понятие: соответствие команды и фактического движения. Первый блок: педагог проверяет питание, подставку, предел времени и физический выключатель; каждый загружает короткую моторную команду с остановкой. Второй блок: смена ролей, изменение одного безопасного параметра, повторный запуск.

Результат: личный запуск и демонстрация остановки; **РУБИ**, колёса вывешены.

28. Почему время не равно расстоянию. Понятие: разброс испытаний. Первый блок: по данным трёх подготовленных коротких проездов сравнить результат и назвать возможные причины различий. Второй блок: на подставке сравнить две краткие длительности включения, останавливая привод между пробами. **Результат:** журнал и вывод о пределах управления по времени; **РУБИ** на подставке.

29. Приоритет безопасности. Понятие: порядок условий. Первый блок: на экранной модели проверить «свободно», «препятствие», «ошибка датчика». Второй блок: написать функцию решения, где препятствие и ошибка всегда дают **stop**; проверить таблицу случаев. **Результат:** правило и три автоматические проверки; **экран**.

30. Проверка правила на устройстве. Понятие: перенос и контроль отказа. Первый блок: на подставке сначала испытать остановку, затем короткое вращение при свободном пути. Второй блок: проверить близкий предмет и ошибку датчика; при обоих колёса стоят. **Результат:** три испытания, личное объяснение и карта пятой проверки; **РУБИ**, колёса вывешены.

Этап 6. Личный проект — занятия 31–36

31. Выбираем миссию. Понятие: требования и минимальная версия. Первый блок: выбрать «настольный сторож», «сигнальщик расстояния» или согласованную идею из проверенных функций. Второй блок: записать «вход → правило → действие», три признака успеха и безопасную реакцию на сбой. **Результат:** личная карточка проекта; **экран**.

32. Первая версия. Понятие: разбиение на функции. Первый блок: собрать правило на записанных данных. Второй блок: отделить получение значения, решение и ответ; проверить обычный и ошибочный случай. **Результат:** личный исходный код версии 0.1; **экран**.

33. Перенос на РУБИ. Понятие: интеграция по частям. Первый блок: отдельно испытать датчик и исполнительный сигнал. Второй блок: каждый загружает личный скетч, проводит одну полную попытку и сохраняет версию.

Результат: работающая миссия; **РУБИ** неподвижный либо на подставке для допущенного моторного проекта.

34. Пять испытаний. Понятие: повторяемость и журнал. Первый блок: зафиксировать условия и ожидаемые ответы пяти проб. Второй блок: лично провести пробы, записать результат и выбрать одну причину для исправления.

Результат: личный журнал пяти попыток, включая проверку сбоя; **РУБИ**.

35. Исправление и повторная проба. Понятие: версия и подтверждение правки. Первый блок: изменить одно правило, сохранить код «до» и «после». Второй блок: обязательно испытать исправленную версию на РУБИ, записать результат и подготовить двухминутное объяснение. **Результат:** подтверждённая правка и инструкция запуска; **РУБИ**.

36. Фестиваль C++ и РУБИ. Понятие: показ и объяснение. Первый блок: контрольный запуск и первая часть личных показов. Второй блок: остальные показы; каждый отвечает на вопрос о своём коде, демонстрирует ошибочный случай или безопасную остановку. **Результат:** исходный код, журнал испытаний и личная защита работающей миссии; **РУБИ**.

6. Как связаны компьютерная модель и контроллер

На компьютере	На РУБИ	Что уточнить
<code>main()</code> выполняет консольную программу	<code>setup()</code> выполняется один раз, <code>loop()</code> повторяется	Сначала проверяем правило в модели, затем переносим его в скетч
<code>std::cin</code> и записанные данные задают вход	Дальномер возвращает измерение или ошибку	Фактические данные могут меняться и требовать проверки
<code>std::cout</code> показывает ответ	Свет и звук дают наблюдаемый ответ	Сигнал имеет допустимую длительность
Файл CSV хранит испытания на компьютере	Контроллер передаёт показания через проверенный вывод	Хранение файлов на Arduino Nano не является обязательным
Экранная модель ограничивает число шагов	Скетч ограничивает время работы и двигательную команду	Физическая остановка проверяется отдельно

Переносится структура решения, а не весь компьютерный исходный файл. Для скетчей педагог предоставляет проверенные настройки платы и назначения выводов; ученики не угадывают их по чужим схемам.

7. Безопасность и доступность

- Перед каждым занятием с РУБИ взрослый проверяет питание, крепления, целостность проводов, свободную площадку, физический выключатель и загруженный стартовый скетч.
- Переподключение питания, аккумуляторов, драйвера и датчиков выполняет взрослый при выключенном устройстве. На перерыве робот переводится в безопасное состояние.
- Новая моторная программа сначала испытывается с вывешенными колёсами и на минимальной допустимой длительности. Движение по полу не входит в обязательный итог курса.
- При неопределённом показании датчика, тайм-ауте или выходе значения за допустимые пределы движение не разрешается. Световой и звуковой ответ также имеют ограниченную длительность.
- Партнёры меняются ролями, но каждый имеет собственный файл, личный запуск и личную запись в журнале. Демонстрация другого ученика не заменяет практику.
- Педагог выдаёт памятки синтаксиса, минимальные рабочие шаблоны и записанные показания для тех, кому нужен более спокойный темп. Усложнение даётся после обязательного результата.

8. Контроль и итоговое задание

После занятий 6, 12, 18, 24, 30 и 36 педагог заполняет краткую карту наблюдения. На первых трёх точках проверяются личный исходный файл, сборка, предсказание результата, тест границы и исправление одной ошибки. На 24-й — личная загрузка скетча, три реакции на данные датчика и безопасное завершение. На 30-й — остановка привода при препятствии и ошибке, личный запуск на подставке. На 36-й — работающий проект и объяснение своего решения. Для каждой графы ставится «сам», «с подсказкой» или «вместе с педагогом» и записывается следующий небольшой шаг.

Итоговое задание: написать личный скетч C++ для РУБИ с реальным входом от дальномера, одним правилом выбора и видимым или слышимым ответом. Обязательны собственная функция, отдельная безопасная реакция на сбой измерения, ограниченное завершение, пять записанных испытаний, одна подтверждённая правка и краткое объяснение кода. Моторы не обязательны; при их использовании проект испытывается только на подставке и лишь на допущенном экземпляре.

Условия зачёта, которые нельзя заменить суммой баллов: личный исходный код и запуск; собственная функция и правило выбора; безопасная реакция на предусмотренный сбой; журнал пяти испытаний и повторная проба после правки. Если условие не выполнено из-за отказа оборудования, педагог предоставляет повторную личную попытку.

Критерий	0	1	2
Личный код C++	Нет работающей версии	Работает с помощью педагога	Ученик изменяет и объясняет функцию и правило
Связь с РУБИ	Нет личного запуска	Запуск с подсказкой	Личный запуск и понятный ответ устройства
Безопасная обработка сбоя	Нет безопасной реакции	Реакция есть с подсказкой	Реакция проверена и объяснена
Испытания и правка	Нет журнала	Журнал неполон или нет повторной пробы	Пять проб, правка и подтверждение результата
Представление	Нет объяснения	Объясняет отдельные части	Называет вход, правило, действие и ограничение

Максимум — 10 баллов. «Зачтено» — от 7 баллов **при выполнении всех обязательных условий**. Недостающий элемент дорабатывается с конкретной обратной связью. Оценочный лист для общего робота заполняется отдельно на каждого участника.

9. Материалы и граница готовности

Педагог готовит 36 карточек миссий, памятку синтаксиса C++, консольные заготовки и записанные наборы данных, проверенные скетчи и схему подключения каждого РУБИ, карту типичных ошибок, шаблон журнала испытаний и отдельные оценочные листы. Для работы с контроллером требуются установленная среда сборки, подтверждённая ревизия платы и тест загрузки на каждый экземпляр.

Этот документ задаёт содержание и результаты курса. Перед первым потоком нужно изготовить и проверить учебные файлы, допустить оборудование к занятиям 19–24 и отдельно к моторным занятиям 27–30. После пилотной группы педагог фиксирует время на личную загрузку каждого участника, частоту отказов оборудования и темы, где требуется дополнительная опора или сокращение усложнений.

Основа адаптации: фактическая конструкция РУБИ, схемы и фотографии используемых экземпляров, существующие материалы по робототехнике с РУБИ. Команды, выводы и параметры переносятся в карточки только после проверки конкретной ревизии. Для синтаксиса C++ педагог использует документацию выбранного компилятора и среды Arduino, установленной на площадке.