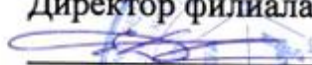


Министерство образования Края
Филиал АНО ДТ «Красноярский «Кванториум» в г. Норильске
«Центр цифрового образования детей IT-Куб г. Норильск»

СОГЛАСОВАНО:

Председатель методического совета
 Т.С.Новгородцева
протокол № 17
от «29» мая 2026 г.

УТВЕРЖДАЮ:

Директор филиала
 Е. А. Дыптан
Приказ № 02-02-92
от «01» 06 2026 г.

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА НА 2026– 2027 уч. год**
технической направленности

**«Микроэлектроника
и программирование на базе платформы Arduino»**

Форма реализации программы – очная
Срок реализации – 1 год
Возраст обучающихся – 14-17 лет

Составитель программы:
Александров Михаил Максимович
педагог дополнительного образования

г. Норильск, 2026 г.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Микроэлектроника и программирование на Arduino» имеет техническую направленность и разработана в соответствии с основными нормативно-правовыми документами: Федеральным Законом «Об образовании» от 29.12.2012 г. № 273-ФЗ; Порядком организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам от 09.11.2018 г. № 196; Целевой моделью развития региональных систем дополнительного образования детей от 03.09.2019 г. № 467; Концепцией развития дополнительного образования детей до 2030 года, утвержденной распоряжением Правительства РФ от 31.03.2022 №678-р; Санитарно-эпидемиологическими требованиями от 28.09.2020 г. № 28.

Уровень программы – стартовый. Освоение программного материала предполагает получение учащимися первоначальных знаний в области микроэлектроники, схемотехники и программирования микроконтроллеров.

Программа ориентирована на обучающихся 12–17 лет, рассчитана на один год в объеме 144 часов.

В настоящее время наиболее перспективной профессиональной средой является сфера embedded-систем, интернета вещей (IoT), робототехники и промышленной автоматизации. Для успешного вхождения в эти области необходимо понимание как электронной схемотехники, так и алгоритмов управления микроконтроллерами.

Программа направлена на становление и развитие начинающих инженеров-схемотехников и программистов микроконтроллеров. Обучающиеся в ходе занятий осваивают принципы работы электронных компонентов, учатся читать электрические схемы, писать код на C++ (Arduino), разрабатывать и защищать собственные проекты. Есть ли наборы?

1.1 НОВИЗНА ДООП

Новизна ДООП «Микроэлектроника и программирование на Arduino: от схемы к устройству» заключается в том, что знакомство обучающихся с основами микроэлектроники и программирования происходит одновременно и во взаимосвязи. В отличие от традиционных курсов, где сначала изучают «чистую» теорию электроники и только потом программирование, в данной программе каждый новый компонент (светодиод, кнопка, датчик, мотор) вводится через практический цикл:

1. Физика работы компонента (доступно, без сложных формул).
2. Сборка схемы на макетной плате с объяснением роли каждого элемента (резистор, конденсатор, транзистор).
3. Написание кода управления на языке C++ в Arduino IDE.
4. Отладка и модификация – изменение параметров, добавление функционала.
5. Мини-проект – интеграция нового компонента в готовое устройство.

1.2 АКТУАЛЬНОСТЬ ПРОГРАММЫ

Актуальность программы обусловлена современным этапом развития общества, характеризующимся тотальной цифровизацией, ускоренными темпами внедрения «умных» устройств (умный дом, носимые гаджеты, промышленная автоматика) и потребностью общества в технически грамотных специалистах начального уровня. Программа полностью отвечает социальному заказу по подготовке кадров для инженерных специальностей.

В Норильске, как крупном промышленном центре, предприятия (Норникель, ТЭЦ, рудники) заинтересованы в притоке молодых людей, понимающих основы автоматизации, электроники и программирования. Ранняя профориентация через практическую работу с Arduino позволяет школьникам сделать осознанный выбор инженерной профессии.

1.3 ПЕДАГОГИЧЕСКАЯ ЦЕЛЕСООБРАЗНОСТЬ

Педагогическая целесообразность программы обусловлена развитием социальных и технических способностей детей через практическое мастерство. Целый ряд специальных заданий на наблюдение, сравнение, поиск неисправностей, домысливание, фантазирование служат для достижения этого.

Педагогическая целесообразность заключается также в раскрытии индивидуальных способностей ребенка, его творческой реализации, формируя качества технически грамотных, трудолюбивых людей, проявляющих интерес к конструированию и программированию. Программа построена так, что каждый обучающийся движется в своём темпе: при выполнении заданий предусмотрены как минимальные требования (базовая схема и код), так и дополнительные задачи для более сильных учащихся (например, добавить второй датчик, изменить алгоритм, добавить энергосбережение).

1.4. ЦЕЛЬ ДООП

Цель программы – сформировать у обучающихся базовые компетенции в области микроэлектроники и программирования микроконтроллеров на платформе Arduino для создания прототипов автоматизированных устройств, а также развить инженерное мышление и навыки проектной работы.

Задачи программы:

1. Познакомить с элементной базой микроэлектроники (резисторы, светодиоды, конденсаторы, транзисторы, датчики, микроконтроллеры).
2. Научить читать простые электрические схемы и собирать их на макетной плате (breadboard).
3. Сформировать навыки программирования на языке C++ в среде Arduino IDE: переменные, типы данных, условные операторы, циклы, функции, работа с библиотеками.
4. Научить подключать и калибровать цифровые (кнопки, PIR) и аналоговые (потенциометр, фоторезистор, LM35) датчики.

5. Обучить управлению исполнительными устройствами: светодиодами, сервоприводами, DC моторами, шаговыми моторами, пьезодинамиками, реле.
6. Сформировать навыки отладки кода и диагностики неисправностей в схеме с помощью мультиметра и последовательного монитора.

1.5. ВОЗРАСТ ОБУЧАЮЩИХСЯ, УЧАСТВУЮЩИХ В ДООП

Программа «Микроэлектроника и её программирование на базе платформы Arduino» рассчитана на обучающихся 12-17 лет. Максимальное количество обучающихся в группе – 12 человек.

1.6. УСЛОВИЯ ВХОЖДЕНИЯ В ДООП

Набор на программу осуществляется в соответствии с Положением о наборе в Филиал АНО ДТ «Красноярский «Кванториум» в г. Норильске.

1.7. СРОК РЕАЛИЗАЦИИ ДООП

Программа рассчитана на 1 учебный год. Нагрузка на обучающегося составляет 144 часа за весь период обучения.

1.8. РЕЖИМ ЗАНЯТИЙ, ФОРМЫ И МЕТОДЫ ОБУЧЕНИЯ

Учебные занятия проходят в очной форме. Режим занятий – 2 раза в неделю по 2 академических часа (1 академический час - 40 минут) с обязательным перерывом.

Виды занятий: лекции-беседы с демонстрацией слайдов и реальных компонентов, практикумы по сборке схем на макетной плате, кодинг-сессии – написание и отладка кода в Arduino IDE, игровые формы работы (например, кто быстрее найдёт ошибку в схеме), соревновательные элементы (чей датчик точнее измеряет расстояние), метод проектов.

1.9. ОЖИДАЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ, СПОСОБЫ ИХ ОТСЛЕЖИВАНИЯ

По результатам обучения обучающиеся овладевают основами микроэлектроники и программирования микроконтроллеров на уровне, достаточном для создания самостоятельных проектов.

Учащиеся будут знать и уметь:

- Назначение основных электронных компонентов (резистор, LED, транзистор, конденсатор, датчики, микроконтроллер;
- Основные команды языка C++ для Arduino (pinMode, digitalWrite, analogRead, analogWrite, delay, millis, if, for, while, serial print, map);
- Правила техники безопасности при работе с электроникой;

Предметные результаты:

- Собирать схемы средней сложности (5–10 компонентов) на breadboard;
- Писать скетчи для управления LED, сервоприводами, моторами, датчиками;
- Использовать последовательный монитор для отладки;

Метапредметные результаты:

- Сформирована готовность к самостоятельному изучению новых датчиков и библиотек;
- Освоен навык изложения технической мысли в чёткой логической последовательности;
- Сформировано умение искать и выделять необходимую информацию в интернете (форумы, документация, примеры кода);

Личностные результаты:

- Знает специфику инженерного труда и технику безопасности;
- Имеет целостный, социально-ориентированный взгляд на окружающий мир (техника вокруг нас);

- Умеет корректно отстаивать свою техническую точку зрения при защите проекта;

Текущий контроль освоения программы проводится во время занятий при помощи опросов, наблюдения за выполнением практических заданий, проведением и поддержанием диалога между учащимися.

Итоговая аттестация по программе проводится в форме письменного, практического задания, защиты проекта.

На основании планируемых результатов разработана оценочная шкала (от 1 до 10 баллов), которая соответствует уровням освоения программы. К концу учебного процесса педагог определяет уровень освоения программы каждым обучающимся.

Низкий уровень (1–3 балла). Характеристика деятельности: обучающийся с трудом собирает простейшую схему (LED + резистор) даже по образцу; допускает грубые ошибки: перепутал землю и питание, замкнул контакты; не знает названий основных компонентов или путает их; в программировании: код не компилируется, не понимает структуру `setup()/loop()`; не может объяснить, как работает собранное устройство. Личностные качества: очень малый словарь технических терминов; не пытается самостоятельно найти ошибку, сразу зовёт педагога; работает без интереса, часто отвлекается.

Средний (допустимый) уровень (4–7 баллов). Характеристика деятельности: обучающийся уверенно собирает схему из 3–5 компонентов по инструкции; знает основные компоненты (LED, резистор, кнопка, Arduino); пишет простые скетчи: мигание, чтение кнопки, потенциометр; есть негрубые ошибки в коде (забыта точка с запятой, неинициализированная переменная), но исправляет их с небольшой подсказкой педагога; может объяснить общий принцип работы своего проекта. Личностные качества: имеет соответствующий программе технический

словарный запас; легко общается с другими учащимися, может помогать им; работу выполняет охотно, но ошибки исправляет только после указания педагога; не всегда проявляет инициативу в усложнении проекта.

Высокий уровень (8–10 баллов). Характеристика деятельности: обучающийся самостоятельно собирает схемы любой сложности в рамках курса, использует документацию; пишет структурированный, хорошо откомментированный код; использует продвинутые конструкции: функции, массивы, `millis()` вместо `delay`, собственные библиотеки; самостоятельно находит и исправляет ошибки (использует `Serial.print`, проверяет соединения мультиметром); активно предлагает собственные идеи для проектов и реализует их. Личностные качества: имеет технический словарный запас выше программы (знает дополнительные термины); сам готов помочь товарищам и объяснить тему; замечает свои ошибки и исправляет их самостоятельно; всегда проявляет фантазию и творчески подходит к созданию устройства; защищает проект уверенно, отвечает на вопросы.

Итоговая оценка выводится как среднее арифметическое по всем контрольным точкам (устные ответы, практические работы, проекты). Уровень фиксируется в журнале и доводится до сведения родителей.

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Название раздела / темы	Всего	Теория	Практика	Промежуточная аттестация / контроль
1	Вводный раздел	2	1	1	Устный опрос
1.1	Цели курса. Техника безопасности. Знакомство с оборудованием. Диагностика стартовых знаний	2	1	1	
2	Основы электроники и Arduino	12	6	6	

2.1	Arduino UNO: плата, микроконтроллер, порты (GND, 5V, digital, analog, PWM, power). Что такое «код» и «прошивка»	2	1	1	
2.2	Breadboard (макетная плата), резисторы (номинал, цветовая маркировка), светодиоды. Закон Ома (качественно). Сборка цепи: LED + резистор	2	1	1	
2.3	Arduino IDE: установка, интерфейс, выбор платы и порта. Структура скетча: setup(), loop(). pinMode(), digitalWrite(), delay()	2	1	1	
2.4	Первая программа: «Мигающий светодиод». Изменение периода мигания. Особенность delay – блокировка	2	1	1	
2.5	Сборка схемы с тремя LED. Написание кода: поочерёдное мигание, бегущий огонёк	2	1	1	
2,6	Обобщение: самостоятельная сборка схемы и написание кода для двух LED разного	2	1	1	Тестирование

	цвета с разным ритмом				
3	Программирование: переменные, ввод-вывод	14	6	8	
3.1	Переменные: int, float, bool. Ввод данных с кнопки: digitalRead, подтяжка INPUT_PULLUP	2	1	1	
3.2	Условный оператор if-else. Управление LED кнопкой: нажал – зажёг, отпустил – погас	2	1	1	
3.3	Аналоговый мир: analogRead, потенциометр. Диапазон 0–1023. Вывод в Serial Monitor	2	1	1	
3.4	analogWrite и ШИМ (PWM) на пинах 3,5,6,9,10,11. Плавное управление яркостью LED	2	1	1	
3.5	Зависимость яркости LED от показаний потенциометра (map() – преобразование диапазонов)	2	1	1	
3.6	Цикл for: повторяющиеся действия. Бегущий огонёк с помощью цикла	2	1	1	
3.7	Контрольная работа: собрать схему «кнопка	2	0	2	Практическая работа

	+ потенциометр + LED». Код: по нажатию кнопки яркость LED задаётся потенциометром				
4	Датчики и исполнительные устройства	20	8	12	
4.1	Подключение сервопривода SG90. Библиотека Servo.h. Управление углом через потенциометр	2	1	1	
4.2	Датчик освещённости (фоторезистор): делитель напряжения. Включаем светодиод при темноте	2	1	1	
4.3	Датчик температуры LM35 (аналоговый) или DHT11 (цифровой). Вывод в Serial	2	1	1	
4.4	Ультразвуковой датчик HC-SR04: trig, echo. Формула расстояния. Вывод на экран	2	1	1	
4.5	Пьезопищалка: tone(), noTone(), частота, длительность. Проигрывание простой мелодии	2	1	1	
4.6	ЖК-дисплей 1602 с I2C адаптером. Подключение (SDA – A4, SCL – A5). Вывод текста и чисел	2	1	1	Самостоятельная практическая работа

4.7	Комбинирование: вывод расстояния с HC-SR04 на LCD	2	1	1	
4.8	Датчик движения PIR (пироэлектрический). Сигнализация: свет + звук	2	1	1	
4.9	Проект №1: «Умный ночник» – светодиод включается по движению или темноте. Выбор логики	4	0	4	Проект
5	Функции и таймеры	12	5	7	
5.1	Пользовательские функции: создание и вызов. void blinkLED(int pin, int duration)	2	1	1	
5.2	Проблема delay – блокировка. Альтернатива: millis() – безблочные таймеры. Мигание LED без остановки кода	2	1	1	
5.3	Массивы: int pins[] = {3,5,6};. Цикл for для работы с массивом пинов	2	1	1	
5.4	Кнопочная матрица (4x4). Подключение, сканирование, вывод нажаткой клавиши	2	1	1	
5.5	Регистр сдвига 74НС595: экономия цифровых пинов.	2	1	1	

	Управление 8 светодиодами через 3 пина				
5.6	Самостоятельная работа: «Бегущий огонёк с регулировкой скорости потенциометром» (без delay)	2	0	2	Контрольная работа
6	Моторы и управление движением	16	6	10	
6.1	DC мотор (коллекторный). Нельзя подключать прямо к Arduino! Транзистор NPN как ключ, защитный диод	2	1	1	
6.2	Драйвер моторов L298N. Питание, управление скоростью (ШИМ) и направлением	2	1	1	
6.3	Управление DC мотором с потенциометра: скорость, направление (две кнопки)	2	1	1	
6.4	Шаговый мотор 28BYJ-48 + драйвер ULN2003. Режимы шага. Вращение на заданный угол	2	1	1	
6.5	Проект №2: «Роботизированная рука» (серва + шаговый мотор) или поворотная платформа	4	1	3	Проект
6.6	Обобщение: сравнение типов моторов	2	1	1	

	(серво, ДС, шаговый) – когда какой приме- нять				
6.7	Тестирование: по опи- санию задачи выбрать мотор и обосновать	2	0	2	Тестирование
7	Беспроводная связь и IoT база	12	6	6	
7.1	Bluetooth модуль HC- 05: режимы (АТ-ко- манды, прозрачный). Сопряжение с телефо- ном	2	1	1	
7.2	Управление LED и сервоприводом через Bluetooth с телефона (простой терминал)	2	1	1	
7.3	Датчик влажности почвы (аналоговый). Порог сухости. Вклю- чение насоса (помпа + реле)	2	1	1	
7.4	Основы I2C и SPI (об- зорно). Зачем нужны. Примеры устройств	2	1	1	
7.5	Проект №3: «Умная теплица» – автомати- ческий полив при су- хости + отправка дан- ных по Bluetooth	4	2	2	Проект
8	Микроэлектроника: продвинутые компо- ненты	14	5	9	

8.1	Транзистор как ключ: NPN, PNP. Управление мощной нагрузкой (лампа 12В через внешний источник)	2	1	1	
8.2	Оптопара и твердотельное реле. Гальваническая развязка. Моделирование безопасного управления	2	1	1	
8.3	Датчик газа MQ-2 (дым, метан). Аналоговый вывод. Порог тревоги	2	1	1	
8.4	Работа с EEPROM (внутренняя память Arduino). Запись и чтение настроек (например, последнее состояние LED)	2	1	1	
8.5	Сдвиговый регистр для ввода: CD4021. Чтение многих кнопок по трём проводам	2	1	1	
8.6	Проект №4: «Охранная система» (PIR + датчик газа + звук + свет + запись тревог в EEPROM)	4	0	4	Проект
9	Программирование: обработка сигналов	12	6	6	
9.1	Фильтрация данных с датчика: медианный фильтр (убираем выбросы).	2	1	1	

9.2	Таймер-прерывание (TimerOne.h). Выполнение кода строго каждые 1 мс независимо от loop	2	1	1	
9.3	Энкодер (инкрементальный). Библиотека Encoder. Определение направления вращения	2	1	1	
9.4	Датчик линии (для роботов): TCRT5000. Чтение чёрного/белого	2	1	1	
9.5	Конечный автомат (state machine) на switch-case. Управление сложным поведением (меню на светодиодах)	2	1	1	
9.6	Чтение технической документации (datasheet) на английском. Находим нужные параметры	2	1	1	
10	Проектный практикум	16	4	12	
10.1	Выбор темы итогового проекта. Разработка технического задания (ТЗ): что делает устройство, какие компоненты	2	1	1	
10.2	Поиск компонентов и библиотек. Составление принципиальной	2	1	1	

	схемы (бумага или онлайн)				
10.3	Сборка схемы прототипа на макетке. Проверка питания, соединений	2	0	2	
10.4	Написание кода: каркас, подключение библиотек, тестирование каждого блока	2	1	1	
10.5	Отладка и интеграция всех подсистем	2	0	2	
10.6	Оформление проекта: схема, код с комментариями, список компонентов, описание работы	2	1	1	
10.7	Предзащита проектов в группе. Взаимное рецензирование	2	0	2	
10.8	Доработка по замечаниям	4	0	4	
11	Итоговый раздел	10	2	8	
11.1	Повторение и обобщение изученного за год: от LED до сложных систем	4	1	3	
11.2	Итоговая аттестация: теоретическая часть (10 вопросов) + практическая (собрать схему и написать код за 1 час)	4	1	3	Аттестация
11.3	Защита итогового проекта (презентация 5–7	4	0	4	Итоговый проект

	минут, демонстрация работы, ответы на вопросы)				
ИТОГО:		144	55	89	

3. СОДЕРЖАНИЕ ПРОГРАММЫ

РАЗДЕЛ 1. ВВОДНЫЙ РАЗДЕЛ (2 часа)

1.1. Цели курса. Техника безопасности. Знакомство с оборудованием. Диагностика стартовых знаний (2 часа)

Теория (1 час): Педагог знакомит обучающихся с программой курса, её целями и задачами. Объясняется, какие устройства научатся создавать дети (умный светильник, охранная система, метеостанция, роботизированная рука и др.). Проводится инструктаж по технике безопасности: правила работы с электронными компонентами, запрет на подключение питания без проверки схемы, правила пайки (только под наблюдением), меры предосторожности при работе с моторами и нагревательными элементами. Демонстрируется оборудование: платы Arduino UNO, макетные платы, наборы резисторов, светодиодов, датчиков, моторов. Объясняется назначение каждого компонента.

Практика (1 час): Проводится входная диагностика: обучающиеся отвечают на вопросы об электричестве (что такое ток, напряжение), пробуют соединить простейшую цепь «батарейка – лампочка» без инструкции, пишут небольшой тест на логическое мышление. Также проверяется навык работы с компьютером (создание папки, сохранение файла). Результаты фиксируются для отслеживания прогресса в конце года.

РАЗДЕЛ 2. ОСНОВЫ ЭЛЕКТРОНИКИ И ARDUINO (12 часов)

2.1. Arduino UNO: плата, микроконтроллер, порты (GND, 5V, digital, analog, PWM, power). Что такое «код» и «прошивка» (2 часа)

Теория (1 час): Строение платы Arduino UNO: микроконтроллер Atmega328P, USB-разъём для питания и прошивки, разъём питания 5–12 В постоянного тока. Подробно разбираются типы портов: земля (GND), питание 5V, цифровые пины (0–13), аналоговые пины (A0–A5), пины с поддержкой ШИМ (3,5,6,9,10,11). Вводится понятие «код» (программа) и «прошивка» (загрузка кода в микроконтроллер). Объясняется разница между микроконтроллером и компьютером.

Практика (1 час): Обучающиеся подключают Arduino к компьютеру через USB-кабель, устанавливают драйверы (при необходимости), запускают Arduino IDE. Определяют номер COM-порта. Проверяют связь через меню «Инструменты → Порт».

2.2. Breadboard (макетная плата), резисторы (номинал, цветовая маркировка), светодиоды. Закон Ома (качественно). Сборка цепи: LED + резистор (2 часа)

Теория (1 час): Устройство беспаячной макетной платы (breadboard): вертикальные и горизонтальные шины питания, ряды контактов. Правила подключения компонентов. Резисторы: назначение (ограничение тока), номинал в Омах, цветовая маркировка (первые три полосы – значение, четвёртая – точность). Светодиоды (LED): анод (длинная ножка) – плюс, катод – минус. Качественное объяснение закона Ома: чем больше сопротивление, тем меньше ток. Почему без резистора светодиод сгорает.

Практика (1 час): Сборка первой цепи на макетной плате: от пина 13 Arduino через резистор 220 Ом к аноду светодиода, катод на GND. Проверка правильности сборки. Включение встроенного примера Blink.

2.3. Arduino IDE: установка, интерфейс, выбор платы и порта. Структура скетча: setup(), loop(). pinMode(), digitalWrite(), delay() (2 часа)

Теория (1 час): Интерфейс Arduino IDE: зона редактора, область вывода сообщений, панель инструментов (верификация, загрузка, новый скетч, сохранение).

Настройка платы (Tools → Board → Arduino Uno) и порта. Структура скетча: функция `setup()` – выполняется один раз при запуске, функция `loop()` – выполняется бесконечно. Команда `pinMode(пин, режим)` – настройка пина на вход (INPUT) или выход (OUTPUT). `digitalWrite(пин, значение)` – подача HIGH (5V) или LOW (0V). `delay(мс)` – пауза в миллисекундах.

Практика (1 час): Обучающиеся пишут первый скетч: мигание светодиодом на пине 13. Компилируют (Verify) и загружают (Upload) на плату. Экспериментируют с изменением задержки `delay()`.

2.4. Первая программа: «Мигающий светодиод». Изменение периода мигания. Особенность `delay` – блокировка (2 часа)

Теория (1 час): Разбор кода мигающего светодиода построчно. Вводится понятие «блокирующая задержка»: во время выполнения `delay()` микроконтроллер не может выполнять другие действия (например, считывать кнопку). Упоминается, что в будущем вместо `delay()` будем использовать `millis()`.

Практика (1 час): Обучающиеся самостоятельно меняют код: мигание с разными интервалами (долго горит – быстро мигает, или ритм «азбука Морзе» для буквы «S»: три коротких, три длинных, три коротких). Педагог проверяет понимание.

2.5. Сборка схемы с тремя LED. Написание кода: поочерёдное мигание, бегущий огонёк (2 часа)

Теория (1 час): Расширение схемы: три светодиода на трёх разных цифровых пинах (например, 9, 10, 11), каждый со своим резистором 220 Ом. Варианты мигания: поочерёдное (1 зажёг – остальные погашены, потом 2 и т.д.) и бегущий огонёк (каждый LED включается по очереди с небольшим перекрытием). Обсуждается логика работы цикла.

Практика (1 час): Сборка схемы с тремя LED. Написание кода для поочерёдного мигания (без цикла for) и для бегущего огонька. Педагог демонстрирует, как упростить код с помощью массива пинов и цикла for (предварительный показ).

2.6. Обобщение: самостоятельная сборка схемы и написание кода для двух LED разного цвета с разным ритмом (2 часа)

Теория (0 часов – практическое занятие): Повторение пройденного материала. Обсуждение типичных ошибок: забыли GND, перепутали анод с катодом, не настроили pinMode, не поставили delay.

Практика (2 часа): Самостоятельная работа: каждый обучающийся собирает схему из двух светодиодов (красный на пине 9, синий на пине 10) и пишет код, в котором красный мигает с частотой 1 Гц (0,5 с горит, 0,5 с не горит), а синий мигает с частотой 0,5 Гц (1 с горит, 1 с не горит). Результат демонстрируется педагогу. Контроль – тестирование (проверка рабочей схемы и кода).

РАЗДЕЛ 3. ПРОГРАММИРОВАНИЕ: ПЕРЕМЕННЫЕ, ВВОД-ВЫВОД (14 часов)

3.1. Переменные: int, float, bool. Ввод данных с кнопки: digitalRead, подтяжка INPUT_PULLUP (2 часа)

Теория (1 час): Понятие переменной – контейнер для данных. Типы данных: int (целые числа), float (дробные), bool (true/false). Объявление переменной, присваивание значения. Подключение тактовой кнопки: две ножки замыкаются при нажатии. Проблема «висящего» входа – решение: подтяжка к питанию через резистор 10 кОм или внутренняя подтяжка INPUT_PULLUP. digitalRead(пин) – возвращает HIGH или LOW.

Практика (1 час): Сборка схемы: кнопка на пине 2 (с внутренней подтяжкой) и светодиод на пине 13. Написание кода: при нажатой кнопке – светодиод горит, при отпущенной – гаснет (или наоборот, с учётом логики INPUT_PULLUP).

3.2. Условный оператор if-else. Управление LED кнопкой: нажал – зажёг, отпустил – погас (2 часа)

Теория (1 час): Синтаксис if-else: if (условие) { действие } else { другое действие }. Операторы сравнения: ==, !=, >, <, >=, <=. Логические операторы: && (и), || (или), ! (не). Переворот логики: кнопка на INPUT_PULLUP даёт LOW при нажатии, поэтому условие if (digitalRead(2) == LOW).

Практика (1 час): Обучающиеся модифицируют код: нажал кнопку – светодиод загорелся и горит до следующего нажатия (триггер). Или: одна кнопка включает, другая выключает. Педагог показывает пример с переменной состояния (boolean lightState).

3.3. Аналоговый мир: analogRead, потенциометр. Диапазон 0–1023. Вывод в Serial Monitor (2 часа)

Теория (1 час): Аналоговый сигнал – непрерывное значение от 0 до 5 В. Функция analogRead(аналоговый_пин) – возвращает число от 0 (0 В) до 1023 (5 В). Потенциометр – переменный резистор, делитель напряжения. Последовательный монитор (Serial.begin(9600), Serial.print(), Serial.println()) для отладки и вывода данных.

Практика (1 час): Сборка схемы: потенциометр (средний вывод на A0, крайние – на 5V и GND). Написание кода, выводящего значение потенциометра в Serial Monitor. Обучающиеся вращают ручку и наблюдают изменение чисел.

3.4. analogWrite и ШИМ (PWM) на пинах 3,5,6,9,10,11. Плавное управление яркостью LED (2 часа)

Теория (1 час): Широтно-импульсная модуляция (ШИМ/PWM): быстрое включение/выключение сигнала, человеческий глаз видит среднюю яркость. Функция

`analogWrite(пин, значение)` где значение от 0 (всегда выкл) до 255 (всегда вкл).
Работает только на пинах с маркировкой ~ (3,5,6,9,10,11).

Практика (1 час): Светодиод на пине 9 через резистор 220 Ом. Код: в `loop()` значение яркости увеличивается от 0 до 255, затем уменьшается (эффект «дыхания»).
Используется `analogWrite` и `delay()`.

3.5. Зависимость яркости LED от показаний потенциометра (`map()` – преобразование диапазонов) (2 часа)

Теория (1 час): Проблема: `analogRead` даёт 0–1023, а `analogWrite` принимает 0–255. Функция `map(значение, fromLow, fromHigh, toLow, toHigh)` – линейное преобразование диапазона. Пример: `brightness = map(sensorValue, 0, 1023, 0, 255);`.
Опционально: функция `constrain()` для ограничения.

Практика (1 час): Сборка схемы: потенциометр на A0, LED на пине 9. Код: считываем потенциометр, преобразуем диапазон, выводим на LED. Обучающиеся вращают ручку – яркость плавно меняется.

3.6. Цикл `for`: повторяющиеся действия. Бегущий огонёк с помощью цикла (2 часа)

Теория (1 час): Цикл `for`: `for (инициализация; условие; инкремент) { тело }`. Пример: `for (int i = 0; i < 10; i++)`. Использование `for` для перебора пинов. Преимущество перед повторяющимся кодом.

Практика (1 час): Сборка схемы с 8 светодиодами на пинах 2–9, каждый с резистором. Написание кода для бегущего огонька: включаем `LED[i]`, задержка, выключаем, переходим к `i+1`. с помощью цикла `for`. Дополнительно: бегущий огонёк в обе стороны.

3.7. Контрольная работа: собрать схему «кнопка + потенциометр + LED».

Код: по нажатию кнопки яркость LED задаётся потенциометром (2 часа)

Теория (0 часов): Только инструктаж перед работой.

Практика (2 часа): Обучающиеся самостоятельно собирают схему: кнопка (на пине 2 с INPUT_PULLUP), потенциометр (на A0), светодиод (на пине 9). Задание: пока кнопка не нажата – LED не горит (или горит на минимальной яркости). После нажатия – устанавливается яркость, равная значению потенциометра (преобразованному в 0–255). Отпускание кнопки не сбрасывает яркость (она сохраняется). Форма контроля: практическая работа с оценкой.

РАЗДЕЛ 4. ДАТЧИКИ И ИСПОЛНИТЕЛЬНЫЕ УСТРОЙСТВА (20 часов)

4.1. Подключение сервопривода SG90. Библиотека Servo.h. Управление углом через потенциометр (2 часа)

Теория (1 час): Сервопривод – устройство с обратной связью, удерживающее заданный угол (обычно 0–180°). Три провода: питание (красный, 5V от Arduino или внешний), земля (чёрный/коричневый), сигнальный (жёлтый/оранжевый). Библиотека Servo.h: `servo.attach(пин)`, `servo.write(угол)`. Управление через потенциометр – считываем аналоговое значение и преобразуем в 0–180.

Практика (1 час): Сборка схемы: сервопривод подключён к 5V, GND и сигнальному пину 9. Потенциометр на A0. Код: `угол поворота сервы = map(analogRead(A0), 0, 1023, 0, 180)`. Обучающиеся вращают потенциометр – серва поворачивается.

4.2. Датчик освещённости (фоторезистор): делитель напряжения. Включаем светодиод при темноте (2 часа)

Теория (1 час): Фоторезистор – меняет сопротивление в зависимости от освещённости (чем светлее, тем ниже сопротивление). Делитель напряжения: фоторезистор + постоянный резистор 10 кОм. Считывание аналогового значения: чем темнее, тем выше значение (или ниже – зависит от схемы). Логика управления: если ниже порога – темно, включаем светодиод.

Практика (1 час): Сборка схемы: фоторезистор и резистор 10 кОм образуют делитель на A0. Светодиод на пине 9. Написание кода: считываем значение, выводим в Serial, задаём порог (например, 500). Если значение ниже порога (темно) – включаем LED, иначе выключаем.

4.3. Датчик температуры LM35 (аналоговый) или DHT11 (цифровой). Вывод в Serial (2 часа)

Теория (1 час): LM35 – аналоговый датчик: 10 мВ на градус Цельсия. Формула: температура = (analogRead(A0) * 5.0 / 1023.0) * 100. DHT11 – цифровой, использует одножичный протокол, требует библиотеку DHT.h, выдаёт температуру и влажность.

Практика (1 час): Обучающиеся по выбору (или по указанию педагога) работают с LM35 или DHT11. Для DHT11: подключение (VCC, GND, сигнал на цифровой пин), установка библиотеки, вывод температуры и влажности в Serial Monitor каждые 2 секунды.

4.4. Ультразвуковой датчик HC-SR04: trig, echo. Формула расстояния. Вывод на экран (2 часа)

Теория (1 час): Принцип работы: излучаем короткий ультразвуковой импульс (trig), засекаем время возвращения эха (echo). Расстояние = (время * скорость звука) / 2. Скорость звука ~343 м/с = 0.0343 см/мкс. Формула: distance = duration * 0.0343 / 2. Подача pulseIn(пин, HIGH) для измерения длительности импульса.

Практика (1 час): Сборка схемы: HC-SR04 – VCC на 5V, GND на GND, Trig на пин 9, Echo на пин 10. Код: запрос измерения, расчёт расстояния, вывод в Serial Monitor. Дополнительно: при расстоянии меньше 20 см включается светодиод (сигнализация приближения).

4.5. Пьезопищалка: tone(), noTone(), частота, длительность. Проигрывание простой мелодии (2 часа)

Теория (1 час): Пьезоизлучатель (пищалка) – создаёт звук при подаче переменного напряжения. Функция `tone(пин, частота, длительность)` – длительность в миллисекундах. `noTone(пин)` – остановка звука. Ноты и их частоты: C4=262 Гц, D4=294, E4=330, F4=349, G4=392, A4=440, B4=494, C5=523. Массив частот и длительностей для мелодии.

Практика (1 час): Подключение пищалки к пину 8 и GND. Написание кода для проигрывания мелодии «В лесу родилась ёлочка» или «Happy Birthday». Использование массива нот и длительностей, цикл `for`.

4.6. ЖК-дисплей 1602 с I2C адаптером. Подключение (SDA – A4, SCL – A5).

Вывод текста и чисел (2 часа)

Теория (1 час): Дисплей 1602 – 16 символов в 2 строках. I2C адаптер использует 4 провода (VCC, GND, SDA, SCL) вместо 16. Подключение: SDA к A4 на Arduino Uno, SCL к A5. Библиотека `LiquidCrystal_I2C.h`. Команды: `lcd.init()`, `lcd.backlight()`, `lcd.setCursor(колонка, строка)`, `lcd.print()`.

Практика (1 час): Подключение дисплея, проверка адреса I2C (скетч сканера). Написание кода: вывод «Hello, world!» на первой строке, счётчик секунд на второй строке (обновляется каждую секунду). Контроль – самостоятельная практическая работа (по желанию педагога).

4.7. Комбинирование: вывод расстояния с HC-SR04 на LCD (2 часа)

Теория (0.5 часа): Объединение двух предыдущих тем. Обсуждается структура кода: в `loop()` измеряем расстояние, затем выводим его на LCD. Форматирование: очистка экрана перед новым выводом или обновление только нужной позиции.

Практика (1.5 часа): Обучающиеся подключают HC-SR04 и LCD одновременно. Пишут код, который каждые 0,5 секунды выводит на экран: «Distance: X cm», где X – текущее расстояние. При расстоянии менее 10 см текст начинает мигать или выводится предупреждение.

4.8. Датчик движения PIR (пирозлектрический). Сигнализация: свет + звук (2 часа)

Теория (1 час): Датчик движения PIR (HC-SR501) реагирует на изменение инфракрасного излучения (тепло человека). Три вывода: VCC, GND, OUT (HIGH при движении). Регулировки: чувствительность и время задержки. Для тестов используем режим однократного срабатывания.

Практика (1 час): Сборка схемы: PIR на цифровой пин 3, светодиод на 9, пин-щетка на 8. Код: при обнаружении движения включается сигнализация (светодиод мигает + звук 1 кГц в течение 3 секунд), затем всё отключается до следующего срабатывания. Дополнительно: кнопка для сброса тревоги.

4.9. Проект №1: «Умный ночник» – светодиод включается по движению или темноте. Выбор логики (4 часа)

Теория (1 час): Постановка технического задания:

- Устройство должно включать светодиод (или ленту), когда в помещении темно И есть движение.
- Альтернативный вариант: ИЛИ (темно ИЛИ движение).
- Обсуждаются варианты датчиков: PIR (движение) + фоторезистор (темнота).
- Логика работы: считываем оба датчика, принимаем решение (if-else с && или ||).

Практика (3 часа): Самостоятельная сборка схемы и написание кода. Обучающиеся могут работать в парах. В конце занятия – презентация проекта (1–2 минуты): демонстрация работы, объяснение схемы и кода. Форма контроля – проект с оценкой по критериям (схема, код, объяснение, работоспособность).

РАЗДЕЛ 5. ФУНКЦИИ И ТАЙМЕРЫ (12 часов)

5.1. Пользовательские функции: создание и вызов. void blinkLED(int pin, int duration) (2 часа)

Теория (1 час): Цель функций – избежать повторения кода. Синтаксис: тип_возврата имя (параметры) { тело }. void – функция ничего не возвращает. Пример: void blinkLED(int pin, int duration) { digitalWrite(pin, HIGH); delay(duration); digitalWrite(pin, LOW); delay(duration); }. Вызов в loop(): blinkLED(13, 500);.

Практика (1 час): Переписать код бегущего огонька с тремя LED, выделив мигание в отдельную функцию. Добавить параметр скорости. Обучающиеся создают свою функцию для плавного изменения яркости.

5.2. Проблема delay – блокировка. Альтернатива: millis() – безблочные таймеры. Мигание LED без остановки кода (2 часа)

Теория (1 час): Проблема delay(): пока идёт задержка, Arduino не может читать кнопку или датчик. Функция millis() – возвращает количество миллисекунд с момента запуска. Принцип безблочного таймера: if (millis() - previousMillis >= interval) { ... previousMillis = millis(); }. Реализация мигающего LED без delay.

Практика (1 час): Написание кода, в котором LED мигает с частотой 1 Гц (через millis), и одновременно в Serial каждые 2 секунды выводится «...». Потенциометр регулирует интервал мигания – всё без задержек.

5.3. Массивы: int pins[] = {3,5,6};. Цикл for для работы с массивом пинов (2 часа)

Теория (1 час): Массив – набор однотипных данных. Объявление: тип имя[размер] = {значения}; или тип имя[] = {значения};. Доступ к элементу: имя[индекс] (индексация с 0). Использование массива пинов и цикла for для управления группой светодиодов.

Практика (1 час): Сборка схемы с 4 светодиодами на пинах 3,5,6,9. Код: бегущий огонёк с помощью цикла `for` и массива. Дополнительно: настройка скорости вращения потенциометром (`map` и `millis`).

5.4. Кнопочная матрица (4x4). Подключение, сканирование, вывод нажатой клавиши (2 часа)

Теория (1 час): Принцип матричной клавиатуры: столбцы – выходы, строки – входы (или наоборот). Поэтапное сканирование: подаём LOW на один столбец, читаем все строки, определяем нажатую клавишу. Готовая библиотека `Keypad.h` упрощает задачу.

Практика (1 час): Подключение матричной клавиатуры 4x4 (8 проводов) к цифровым пинам 2–9. Установка библиотеки `Keypad`. Написание кода: при нажатии клавиши выводить её в `Serial Monitor` и на LCD (если подключен). Реализация простого калькулятора (сложение двух чисел).

5.5. Регистр сдвига 74HC595: экономия цифровых пинов. Управление 8 светодиодами через 3 пина (2 часа)

Теория (1 час): 74HC595 – сдвиговый регистр с защёлкой. Позволяет управлять 8 выходами через 3 пина Arduino: `data` (DS), `clock` (SH_CP), `latch` (ST_CP). Последовательная передача байта: `shiftOut(dataPin, clockPin, LSBFIRST, value)`; затем `latch`-импульс.

Практика (1 час): Сборка схемы: 74HC595, 8 светодиодов с резисторами. Подключение: DS к 11, SH_CP к 13, ST_CP к 10. Код: зажигаем светодиоды по очереди (бегущий огонёк) с помощью цикла `for` и битовых масок.

5.6. Самостоятельная работа: «Бегущий огонёк с регулировкой скорости потенциометром» (без `delay`) (2 часа)

Теория (0 часов): Повторение `usecase`: безблочный таймер + массив пинов + потенциометр.

Практика (2 часа): Задание: собрать схему с 8 светодиодами через 74НС595 (или напрямую на пинах). Написать код, реализующий бегущий огонёк (светодиод сдвигается каждые N мс). Скорость движения (интервал) регулируется потенциометром. Использовать millis(), без delay. Форма контроля – контрольная работа (чек-лист работоспособности).

РАЗДЕЛ 6. МОТОРЫ И УПРАВЛЕНИЕ ДВИЖЕНИЕМ (16 часов)

6.1. DC мотор (коллекторный). Нельзя подключать прямо к Arduino! Транзистор NPN как ключ, защитный диод (2 часа)

Теория (1 час): DC мотор потребляет ток 100–500 мА, что превышает допустимый для выхода Arduino (40 мА). Схема управления: транзистор NPN (2N2222, BC547) в ключевом режиме. База через резистор 1 кОм к пину Arduino, коллектор к минусу мотора, эмиттер на GND. Плюс мотора – к +5В (или внешнему питанию). Диод 1N4007 параллельно мотору (анод к коллектору) – защита от обратной ЭДС.

Практика (1 час): Сборка схемы на макетной плате. Написание кода: высокий уровень на управляющем пине – мотор вращается, низкий – останавливается. Измерение тока мультиметром.

6.2. Драйвер моторов L298N. Питание, управление скоростью (ШИМ) и направлением (2 часа)

Теория (1 час): L298N – H-мост, позволяет управлять двумя моторами (скорость и направление). Питание логики 5V (от Arduino или регулятора на плате), питание моторов до 12В отдельно. Входы: IN1, IN2 для мотора А (IN1=HIGH, IN2=LOW – вращение в одну сторону, наоборот – в другую). EnableA (PWM) – регулировка скорости.

Практика (1 час): Подключение одного DC мотора к L298N (OUT1, OUT2). IN1 к пину 4, IN2 к пину 5, ENA к пину 9 (PWM). Написание кода: потенциометр задаёт скорость (0–255), кнопка меняет направление.

6.3. Управление DC мотором с потенциометра: скорость, направление (две кнопки) (2 часа)

Теория (0.5 часа): Разбор логики: потенциометр – скорость, две кнопки – «вперёд» и «назад». При нажатии «вперёд» – IN1=HIGH, IN2=LOW; «назад» – наоборот. Если ни одна кнопка не нажата – останов (оба LOW). Скорость через analogWrite на ENA.

Практика (1.5 часа): Сборка полной схемы (L298N + мотор, потенциометр на A0, кнопки на пинах 2 и 3). Написание кода с учётомдребезга кнопок (простая задержка 50 мс). Проверка работы.

6.4. Шаговый мотор 28BYJ-48 + драйвер ULN2003. Режимы шага. Вращение на заданный угол (2 часа)

Теория (1 час): Шаговый мотор вращается дискретно (шагами). 28BYJ-48 имеет 2048 шагов на оборот (при full-step). Драйвер ULN2003 имеет 4 входа (IN1–IN4). Библиотека Stepper.h или AccelStepper.h. Команды: setSpeed(), step().

Практика (1 час): Подключение: IN1–IN4 к пинам 8–11, питание мотора от 5V или 9V (внешнее). Код: вращение на 2048 шагов (один оборот) в одну сторону, задержка, затем в другую. Дополнительно: вращение на угол 90° (512 шагов).

6.5. Проект №2: «Роботизированная рука» (серва + шаговый мотор) или поворотная платформа (4 часа)

Теория (1 час): Постановка задачи: создать устройство, которое поворачивается на шаговом моторе (основание) и поднимает/опускает «руку» на сервоприводе. Управление: два потенциометра (один для угла поворота основания, другой для угла руки). Обсуждение механики.

Практика (3 часа): Сборка схемы (шаговый мотор + драйвер, сервопривод).

Написание кода: считывание двух потенциометров, преобразование одного в угол для сервы (0–180), другого – в количество шагов для шагового мотора (0–2048).

Форма контроля – проект с демонстрацией.

6.6. Обобщение: сравнение типов моторов (серво, DC, шаговый) – когда какой применять (2 часа)

Теория (1 час): Таблица характеристик: точность позиционирования, необходимость обратной связи, стоимость, сложность управления. Серва – для позиционирования под нагрузкой (угол). Шаговый – для точного позиционирования без обратной связи (3D-принтеры, станки). DC мотор – для вращения с регулировкой скорости (колёса робота, вентиляторы).

Практика (0 часов): Дискуссия, решение кейсов (педагог называет задачу – обучающиеся выбирают мотор).

6.7. Тестирование: по описанию задачи выбрать мотор и обосновать (2 часа)

Теория (0 часов): Только инструктаж.

Практика (2 часа): Письменный тест из 10 вопросов-кейсов. Примеры: «Робот-пылесос должен объезжать препятствия и поворачиваться на месте – какой мотор для колёс?», «Механизм открывания штор – серва или шаговый?». Форма контроля – тестирование.

РАЗДЕЛ 7. БЕСПРОВОДНАЯ СВЯЗЬ И ИОТ БАЗА (12 часов)

7.1. Bluetooth модуль HC-05: режимы (AT-команды, прозрачный). Сопряжение с телефоном (2 часа)

Теория (1 час): HC-05 – последовательный Bluetooth-адаптер. Два режима: AT-команды (настройка имени, скорости, роли) и прозрачный (передача данных как через Serial). Подключение: TXD к RX (пин 0), RXD к TX (пин 1) через делитель

напряжения (3.3V логика!). Важно: при прошивке Arduino отсоединять RX/TX. Сопряжение: пароль 1234 или 0000.

Практика (1 час): Подключение HC-05 к Arduino (питание 5V, TX->RX, RX->TX через делитель). Установка последовательной связи SoftwareSerial на других пинах (например, 10, 11). Отправка и приём простых сообщений с телефона (приложение Serial Bluetooth Terminal).

7.2. Управление LED и сервоприводом через Bluetooth с телефона (простой терминал) (2 часа)

Теория (0.5 часа): Протокол: отправляем с телефона символы: '1' – LED вкл, '0' – выкл, 'L' – включить серву влево, 'R' – вправо. В Arduino: Serial.read() или readString(). Разбор команды через switch-case.

Практика (1.5 часа): Сборка: LED на пине 9, серва на пине 10, HC-05 на пинах 4(RX) и 5(TX) через SoftwareSerial. Написание кода, который принимает команды и выполняет действия. Обучающиеся управляют устройством с телефона.

7.3. Датчик влажности почвы (аналоговый). Порог сухости. Включение насоса (помпа + реле) (2 часа)

Теория (1 час): Датчик влажности почвы (два электрода) – выводит аналоговое значение: чем суше, тем выше значение (или наоборот). Порог сухости подбирается экспериментально. Включение насоса (помпы) через реле (модуль реле с оптопарой). Важно: реле управляет внешним питанием 12В для насоса.

Практика (1 час): Сборка схемы: датчик влажности на A0, модуль реле (управляющий пин на 7). Код: считываем влажность, если значение выше порога (сухо) – включаем насос на 2 секунды, затем выключаем на 10 минут (millis). Вывод в Serial.

7.4. Основы I2C и SPI (обзорно). Зачем нужны. Примеры устройств (2 часа)

Теория (1 час): Проблема ограниченного числа пинов. I2C (Inter-Integrated Circuit): два провода (SDA, SCL), множество устройств с разными адресами. Примеры: дисплей 1602 (I2C версия), датчик BME280. SPI (Serial Peripheral Interface): 4 провода (MOSI, MISO, SCK, SS), выше скорость. Примеры: SD-карта, Ethernet-модуль.

Практика (1 час): Подключение двух I2C-устройств к одной шине (например, LCD и RTC часы). Сканирование адресов I2C (скетч-сканер). Краткое знакомство с SPI на примере сдвигового регистра (входного).

7.5. Проект №3: «Умная теплица» – автоматический полив при сухости + отправка данных по Bluetooth (4 часа)

Теория (1 час): Техническое задание: устройство измеряет влажность почвы каждые 5 секунд, если сухо – включает насос на 2 секунды. Также по запросу с телефона (через Bluetooth) отправляет текущую влажность и температуру (если добавлен DHT11). Все параметры (порог сухости, время полива) хранятся в EEPROM.

Практика (3 часа): Самостоятельная сборка и программирование. Обязательные компоненты: датчик влажности, реле + насос, Bluetooth HC-05, опционально DHT11. Форма контроля – проект с обязательной демонстрацией и пояснением кода.

РАЗДЕЛ 8. МИКРОЭЛЕКТРОНИКА: ПРОДВИНУТЫЕ КОМПОНЕНТЫ (14 часов)

8.1. Транзистор как ключ: NPN, PNP. Управление мощной нагрузкой (лампа 12В через внешний источник) (2 часа)

Теория (1 час): Расширение темы 6.1: NPN-транзистор открывается положительным напряжением на базе. PNP – открывается при низком уровне. Для управления нагрузкой выше 5V используется внешний источник. Схема: от Arduino через

резистор 1-10 кОм на базу, коллектор – к нагрузке, нагрузка – к +питания, эмиттер – на GND. Диод для защиты.

Практика (1 час): Сборка схемы управления лампой 12В (или мощным светодиодом) от Arduino через транзистор. Внешний источник 12В. Код: вкл/выкл по кнопке. Измерение тока и напряжения.

8.2. Оптопара и твердотельное реле. Гальваническая развязка. Моделирование безопасного управления (2 часа)

Теория (1 час): Оптопара (например, PC817) – светодиод и фототранзистор в одном корпусе, развязка цепей. Применение: управление высоким напряжением (220В) безопасно для Arduino. Твердотельное реле (SSR) – готовая сборка на оптопаре и симисторе.

Практика (1 час): Безопасное моделирование: управление светодиодом через оптопару. Arduino включает светодиод оптопары, фотоприёмник открывает транзистор, который включает внешний светодиод (от отдельной батарейки). Измерение изоляции (нет гальванической связи).

8.3. Датчик газа MQ-2 (дым, метан). Аналоговый вывод. Порог тревоги (2 часа)

Теория (1 час): Датчики серии MQ требуют прогрева (несколько минут). MQ-2 чувствителен к дыму, метану, пропану. Аналоговый выход даёт напряжение, пропорциональное концентрации газа. Цифровой выход (DO) срабатывает при превышении порога (настраивается потенциометром на модуле).

Практика (1 час): Подключение MQ-2 (питание 5V, AOUT на A0). Код: считываем значение, выводим в Serial. Подбираем порог срабатывания для зажигалки (без пламени) или ватных палочек с дымом. При превышении порога включаем светодиод и пищалку.

8.4. Работа с EEPROM (внутренняя память Arduino). Запись и чтение настроек (например, последнее состояние LED) (2 часа)

Теория (1 час): EEPROM (Electrically Erasable Programmable Read-Only Memory) – 1024 байта на Arduino UNO. Не стирается при выключении питания. Функции: EEPROM.read(адрес), EEPROM.write(адрес, значение). Адреса от 0 до 1023. Значение – байт (0–255). Сохранение состояния светодиода: при запуске считываем из EEPROM и устанавливаем LED соответственно.

Практика (1 час): Код: кнопка переключает состояние светодиода и сохраняет новое состояние в EEPROM. После перезагрузки Arduino светодиод восстанавливает последнее состояние. Дополнительно: сохранение порога для датчика освещённости.

8.5. Сдвиговый регистр для ввода: CD4021. Чтение многих кнопок по трём проводам (2 часа)

Теория (1 час): Симметрия 74HC595: CD4021 позволяет читать 8 входов (например, кнопок) через 3 провода (data, clock, parallel load). Принцип: защёлкиваем состояние параллельных входов (load pulse), затем последовательно сдвигаем биты.

Практика (1 час): Сборка схемы: 8 кнопок на параллельные входы CD4021. Выводы data, clock, load к пинам Arduino. Написание кода, который считывает состояние всех 8 кнопок и выводит номер нажатой в Serial.

8.6. Проект №4: «Охранный система» (PIR + датчик газа + звук + свет + запись тревог в EEPROM) (4 часа)

Теория (0 часов): Самостоятельная интеграция.

Практика (4 часа): Разработка комплексного проекта:

- Датчик движения PIR (вход).
- Датчик газа MQ-2 (концентрация).

- При срабатывании любого датчика: включается сирена (пищалка + мигающий светодиод/лампа) на 10 секунд.
- Каждая тревога записывается в EEPROM (номер события и время в секундах от запуска).
- По Bluetooth (или кнопке) можно вывести журнал тревог на LCD.
- Возможность сброса тревог кнопкой.

Форма контроля – защита проекта.

РАЗДЕЛ 9. ПРОГРАММИРОВАНИЕ: ОБРАБОТКА СИГНАЛОВ (12 часов)

9.1. Фильтрация данных с датчика: медианный фильтр (убираем выбросы) (2 часа)

Теория (1 час): Датчики (особенно HC-SR04 и фоторезистор) дают шумные показания. Медианный фильтр: берём N измерений (например, 5), сортируем, берём среднее. Код: массив из 5 значений, цикл сортировки пузырьком, медиана = значения[2].

Практика (1 час): Применение медианного фильтра к ультразвуковому датчику. Сравнение сырых данных и отфильтрованных – график в Serial Plotter (инструмент Arduino IDE). Обучающиеся видят уменьшение выбросов.

9.2. Таймер-прерывание (TimerOne.h). Выполнение кода строго каждые 1 мс независимо от loop (2 часа)

Теория (1 час): Прерывания – остановка основного кода для выполнения срочной задачи. Библиотека TimerOne.h для Arduino. Пример: `Timer1.initialize(1000);` – каждые 1000 мкс (1 мс) вызывать обработчик. `Timer1.attachInterrupt(handler);`. В обработчике нельзя использовать `delay` и `Serial.print` (только простые переменные).

Практика (1 час): Код: прерывание каждые 1 мс инкрементирует переменную counter. В loop() выводим counter каждую секунду (через millis). Наблюдаем точность. Управление сервоприводом через прерывание.

9.3. Энкодер (инкрементальный). Библиотека Encoder. Определение направления вращения (2 часа)

Теория (1 час): Инкрементальный энкодер выдаёт два сигнала (А и В) со сдвигом фазы для определения направления. Библиотека Encoder упрощает работу. Подключение: два цифровых пина (лучше с прерываниями, например 2 и 3).

Практика (1 час): Подключение энкодера с кнопкой (модуль KY-040). Код: вывод текущего счётчика (число импульсов) в Serial и на LCD. Нажатие кнопки обнуляет счётчик.

9.4. Датчик линии (для роботов): TCRT5000. Чтение чёрного/белого (2 часа)

Теория (1 час): TCRT5000 – инфракрасный светодиод + фототранзистор. На чёрной поверхности ИК-луч поглощается, выход LOW (0); на белой – отражается, выход HIGH (1). Использование для роботов, следующих по линии.

Практика (1 час): Подключение (VCC, GND, OUT на цифровой пин). Код: считываем значение, включаем светодиод, если видит белое. Калибровка на разных поверхностях.

9.5. Конечный автомат (state machine) на switch-case. Управление сложным поведением (меню на светодиодах) (2 часа)

Теория (1 час): Конечный автомат – модель с несколькими состояниями и переходами между ними по событиям. Реализация: переменная state (int), в loop() switch(state) { case 0: ... case 1: ... }. Пример: состояние 0 – ожидание кнопки, состояние 1 – мигание, состояние 2 – бегущий огонёк.

Практика (1 час): Код: три светодиода, одна кнопка. Каждое нажатие кнопки меняет режим работы светодиодов (выкл, мигание 1 Гц, бегущий огонёк). Реализовать без delay, только millis и автомат состояний.

9.6. Чтение технической документации (datasheet) на английском. Находим нужные параметры (2 часа)

Теория (1 час): Структура datasheet: общая информация, максимальные электрические параметры, назначение выводов, временные диаграммы, примеры схем. Зачем уметь читать: подбор резисторов, совместимость напряжений, поиск библиотек.

Практика (1 час): Раздаются datasheet на русском/английском для компонентов: LM35, HC-SR04, 74HC595. Задание: найти максимальное напряжение питания, выходной ток, назначение каждого вывода. Ответы записать в тетрадь.

РАЗДЕЛ 10. ПРОЕКТНЫЙ ПРАКТИКУМ (16 часов)

10.1. Выбор темы итогового проекта. Разработка технического задания (ТЗ): что делает устройство, какие компоненты (2 часа)

Теория (1 час): Мозговой штурм: примеры тем – метеостанция (температура, влажность, давление, облачность с помощью датчиков), умная кормушка для кота (вес корма + расписание), индикатор уровня шума, робот-пылесос на DC моторах и датчиках линии, система доступа по кнопочной матрице, музыкальный синтезатор. Как составить ТЗ: цель, список компонентов, алгоритм работы, интерфейс пользователя.

Практика (1 час): Каждый обучающийся (или группа из 2-3 человек) выбирает тему, пишет краткое ТЗ (1 страница), утверждает у педагога.

10.2. Поиск компонентов и библиотек. Составление принципиальной схемы (бумага или онлайн) (2 часа)

Теория (1 час): Обзор онлайн-инструментов: Fritzing, EasyEDA, схемы на бумаге. Как выбирать библиотеки (проверка даты обновления, звёзд на GitHub, примеров). Правила поиска: поиск по запросу «Arduino [модуль] library».

Практика (1 час): Составление полной принципиальной схемы проекта. Список компонентов. Поиск и установка необходимых библиотек.

10.3. Сборка схемы прототипа на макетке. Проверка питания, соединений (2 часа)

Теория (0 часов): Только инструктаж.

Практика (2 часа): Сборка всех компонентов на одной или нескольких макетных платах. Проверка питания (мультиметром), целостности соединений. Поэтапное тестирование: сначала один датчик, потом второй и т.д.

10.4. Написание кода: каркас, подключение библиотек, тестирование каждого блока (2 часа)

Теория (1 час): Иерархия кода: глобальные переменные, `setup()`, `loop()`, вспомогательные функции. Комментирование кода. Прототипирование – написать заглушки.

Практика (1 час): Написание кода по блокам: отдельно считывание датчика, отдельно управление мотором, отдельно вывод на LCD. Тестирование каждого блока перед объединением.

10.5. Отладка и интеграция всех подсистем (2 часа)

Теория (0 часов): Советы по отладке: больше `Serial.print()`, разделение на этапы, проверка питания.

Практика (2 часа): Объединение всех блоков в один скетч. Решение конфликтов пинов (если используется один и тот же пин двумя библиотеками). Оптимизация таймингов.

10.6. Оформление проекта: схема, код с комментариями, список компонентов, описание работы (2 часа)

Теория (1 час): Требования к оформлению: текст описания (1-2 страницы), схема (скриншот или фото макетки), код с комментариями, список компонентов с ценами/артикулами. Подготовка презентации (3-5 слайдов).

Практика (1 час): Оформление проекта в электронном виде. Подготовка речи для защиты (2-3 минуты).

10.7. Предзащита проектов в группе. Взаимное рецензирование (2 часа)

Теория (0 часов): Инструктаж по рецензированию: отметить сильные стороны, указать на ошибки конструктивно.

Практика (2 часа): Каждый обучающийся/группа представляет свой проект (демонстрация работы + краткий рассказ). Другие обучающиеся задают вопросы, педагог даёт рекомендации. Форма контроля – промежуточная оценка.

10.8. Доработка по замечаниям (4 часа)

Теория (0 часов): Индивидуальные консультации.

Практика (4 часа): Устранение замечаний: исправление ошибок в схеме, оптимизация кода, улучшение презентации. Дополнительная консультация с педагогом.

РАЗДЕЛ 11. ИТОГОВЫЙ РАЗДЕЛ (10 часов)

11.1. Повторение и обобщение изученного за год: от LED до сложных систем (4 часа)

Теория (1 час): Структурирование знаний по трём направлениям:

1. Электроника (резисторы, транзисторы, датчики, моторы).
2. Программирование (переменные, условия, циклы, функции, millis, массивы).

3. Инструменты (IDE, Serial Monitor, библиотеки).

Практика (3 часа): Решение комплексных задач: «Спроектируйте систему автоматического проветривания теплицы», «Как бы вы реализовали управление шаговым мотором без библиотеки?». Решение кроссвордов по терминологии.

11.2. Итоговая аттестация: теоретическая часть (10 вопросов) + практическая (собрать схему и написать код за 1 час) (4 часа)

Теория (1 час): Инструктаж, ответы на вопросы.

Практика (3 часа): Итоговая аттестация состоит из двух частей:

- Письменная: 10 вопросов с выбором ответа и открытых вопросов (например, «Объясните разницу между digitalWrite и analogWrite», «Зачем нужен диод параллельно DC мотору?»).
- Практическая: за 1 час участник собирает схему и пишет код для одной из предложенных простых задач (например, «Подключить потенциометр и 3 светодиода. При повороте потенциометра загорается соответствующее количество светодиодов: 0–341 – 1 светодиод, 342–682 – 2 светодиода, 683–1023 – 3 светодиода»).

Форма контроля – итоговая аттестация с оценкой по 10-балльной шкале.

11.3. Защита итогового проекта (презентация 5–7 минут, демонстрация работы, ответы на вопросы) (4 часа)

Теория (0 часов): Регламент защиты.

Практика (4 часа): Публичная защита итоговых проектов перед комиссией (педагог + приглашённые родители или наставники). Критерии оценки: работоспособность устройства (30%), качество кода и схемы (30%), умение объяснить (20%), оригинальность идеи (10%), оформление (10%). Форма контроля – итоговый проект с выставлением оценки в протокол.

4. МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ПРО- ГРАММЕ

- **Оборудование:** компьютерный класс с Windows/Linux, проектор, доступ в интернет.
- **ПО:** Arduino IDE (свободное), библиотеки: DHT, Servo, LiquidCrystal_I2C, NewPing, Encoder.
- **Оборудование на группу 12 человек (комплекты):**
 - Arduino UNO (или совместимый) – 12 шт.
 - Макетная плата breadboard 400 точек – 12 шт.
 - Набор проводов «папа-папа», «папа-мама»
 - Резисторы (220 Ом, 1 кОм, 10 кОм – по 50 шт)
 - Светодиоды 5 мм (красные, зелёные, синие по 30 шт)
 - Кнопки тактовые (20 шт)
 - Потенциометры 10 кОм (12 шт)
 - Ультразвуковой датчик HC-SR04 (12 шт)
 - Сервопривод SG90 (12 шт)
 - Датчик температуры/влажности DHT11 (12 шт)
 - ЖК-дисплей 1602 с I2C (6 шт, общий)
 - Модуль реле (6 шт)
 - Датчик движения PIR (6 шт)
 - Bluetooth HC-05 (6 шт)
 - Шаговый мотор 28BYJ-48 + драйвер (6 шт)
 - DC мотор + транзистор + диод (12 компл.)
 - Регистр сдвига 74HC595 (12 шт)
- **Расходные материалы:** USB-кабели, батарейные отсеки 6В, мультиметры (3 шт.), набор отвёрток.

5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная:

1. Джереми Блум. «Изучаем Arduino: инструменты и методы технического волшебства». – BHV, 2016.

2. Саймон Монк. «Программируем Arduino. Основы работы со скетчами». – Питер, 2018.
3. М. Шмидт. «Arduino: быстрое обучение». – Эксмо, 2015.

Электронные ресурсы:

1. arduino.cc/reference (официальная документация)
2. arduino.ru
3. instructables.com – раздел Arduino

6. СВЕДЕНИЯ О СОСТАВИТЕЛЯХ ДООП

Александров Михаил Максимович – педагог дополнительного образования по направлению «UNITY». Образование неполное высшее.

7. МАТЕРИАЛЫ ДЛЯ ИТОГОВОГО КОНТРОЛЯ

1. Собрать схему: подключение сервопривода к пину 9 и потенциометра к А0. Написать код: угол сервопривода равен значению потенциометра (map).
2. Устно описать алгоритм работы ультразвукового датчика.
3. Найти ошибку в предоставленном неправильном коде (например, забыт pinMode).
4. Представить итоговый проект: демонстрация работы устройства, объяснение схемы и кода, ответы на вопросы по выбору компонентов.
5. Написать короткую аннотацию (техническое описание) своего проекта.