

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САХАЛИНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

**для студентов по выполнению
практических работ
по дисциплине**

ПМ.01 УЧАСТИЕ В ПРОЕКТИРОВАНИИ АРХИТЕКТУРЫ ИНТЕЛЛЕКТУАЛЬНЫХ ИНТЕГРИРОВАННЫХ СИСТЕМ

**укрупненная группа:
09.00.00 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
специальность: 09.02.08 Интеллектуальные
интегрированные системы**

базовый уровень подготовки

**Квалификация: Техник по интеллектуальным интегрированным системам
Форма обучения: очная**

Южно-Сахалинск
2025

УТВЕРЖДАЮ
Зам. директора по НМР
 Е.Н. Ермолаева
«24» марта 2025 г.

Разработчик: Перов И.А., преподаватель информационных дисциплин

Одобрено на заседании ПЦК
информационных дисциплин
Протокол № 8 от «21» марта 2025г.
Председатель ПЦК



О.Б. Куприяшина

СОДЕРЖАНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ	2
2. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К ПРАКТИЧЕСКИМ РАБОТАМ (РАЗДЕЛ 1. ЦИФРОВАЯ СХЕМОТЕХНИКА)	3
Практическая работа №1.....	3
Практическая работа №2.....	3
Практическая работа №3.....	4
Практическая работа №4.....	5
Практическая работа №5.....	6
Практическая работа №6.....	6
Практическая работа №7.....	7
Практическая работа №8.....	7
Практическая работа №9.....	8
Практическая работа №10.....	8
Практическая работа №11.....	9
Практическая работа №12.....	10
Практическая работа №13.....	10
3. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К ПРАКТИЧЕСКИМ РАБОТАМ (РАЗДЕЛ 2. МИКРОКОНТРОЛЛЕРНЫЕ СИСТЕМЫ)	12
Практическая работа №1.....	12
Практическая работа №2.....	12
Практическая работа №3.....	13
Практическая работа №4.....	14
Практическая работа №5.....	14
Практическая работа №6.....	15
4. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К УЧЕБНОЙ ПРАКТИКЕ (72 часа)	17
Вид работы 1. Использование АЦП микроконтроллера	17
Вид работы 2. Взаимодействие с встроенной памятью EEPROM.....	17
Вид работы 3. Взаимодействие со светодиодной матрицей (например, 8×8 MAX7219)	17
Вид работы 4. Взаимодействие с ЖКИ (символьный дисплей 16×2, I2C или параллельный).....	17
Вид работы 5. Работа с цифровым температурным датчиком (DS18B20, 1-Wire).....	17
Вид работы 6. Работа с двигателем постоянного тока (через драйвер L298N или транзистор)	17
Вид работы 7. Работа с серводвигателем (SG90 или MG995)	18
Вид работы 8. Работа с шаговым двигателем (28BYJ-48 с драйвером ULN2003).....	18
Вид работы 9. Работа с модулем передачи информации (например, Bluetooth HC-05 или LoRa, Wi-Fi ESP8266).....	18
5. КРИТЕРИИ ОЦЕНКИ ПРАКТИЧЕСКИХ РАБОТ	19
6. ЛИТЕРАТУРА И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ.....	20

1. ОБЩИЕ ПОЛОЖЕНИЯ

Цели практических работ:

- Закрепить теоретические знания по цифровой схемотехнике и микроконтроллерным системам.
- Сформировать умения собирать, настраивать и тестировать цифровые устройства и системы на базе микроконтроллеров.
- Развить навыки работы с измерительным оборудованием (осциллограф, мультиметр), симуляторами и средами разработки.

Организация выполнения:

- Каждая практическая работа рассчитана на **2–4 академических часа** (согласно РП).
- Выполняется в лаборатории «Электротехники и электроники» или мастерской «Аппаратной инфраструктуры Интернета вещей».
- Работа может выполняться индивидуально или в бригадах по 2 человека.
- Результатом является **отчёт** (краткий протокол)

и демонстрация работающего устройства преподавателю.

Оборудование и ПО (общее):

- Для схемотехники: макетные платы, набор логических микросхем (серии 74НС, К155, К561), источник питания 5 В, осциллограф, мультиметр, логический анализатор (по желанию). Либо симуляторы: **Logisim, Multisim, Proteus**.
- Для микроконтроллеров: плата Arduino Uno / Nano, STM32F103 Blue Pill, или отечественные аналоги; среда Arduino IDE, PlatformIO, Atmel Studio; набор датчиков и исполнительных устройств.

Техника безопасности:

- Перед включением схемы проверить правильность монтажа.
- Не превышать напряжение питания (5 В для ТТЛ/КМОП, 3.3 В для некоторых МК).
- При работе с осциллографом – заземлить приборы.
- После окончания работы отключить питание.

2. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К ПРАКТИЧЕСКИМ РАБОТАМ

(РАЗДЕЛ 1. ЦИФРОВАЯ СХЕМОТЕХНИКА)

Практическая работа №1

«Цифровые логические элементы» (2 часа)

Цель: изучить работу базовых логических вентилях (И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ, исключающее ИЛИ) и снять их передаточные характеристики.

Оборудование: макетная плата, микросхемы 74НС08 (И), 74НС32 (ИЛИ), 74НС04 (НЕ), 74НС00 (И-НЕ), 74НС02 (ИЛИ-НЕ), 74НС86 (XOR); источник 5 В; мультиметр; осциллограф (опционально).

Задание:

1. Для каждого типа вентиля собрать схему и заполнить таблицу истинности (все комбинации входов).
2. Измерить выходное напряжение для каждого состояния, сравнить с логическими уровнями (0 – <0.8 В, 1 – >2.4 В для ТТЛ).
3. С помощью осциллографа (или мультиметра) оценить задержку распространения сигнала (подать меандр 1 кГц, измерить сдвиг между входом и выходом).
4. Сделать вывод о быстродействии и помехоустойчивости.

Порядок выполнения:

- Установить микросхему на макетку, подключить питание (VCC – +5V, GND – 0V).
- Для И-НЕ (74НС00): соединить входы А и В с переключателями, выход – со светодиодом (через резистор 330 Ом) или мультиметром.
- Перебирая 4 комбинации (00,01,10,11), записать выходное напряжение.
- Повторить для других вентилях.

Форма отчёта:

Таблица истинности для каждого вентиля + измеренные напряжения. Осциллограммы (зарисовать). Вывод о соответствии стандартным логическим уровням.

Контрольные вопросы:

1. Чем отличается TTL от CMOS по уровням и потреблению?
2. Почему входы неиспользуемых вентилях нельзя оставлять «в воздухе»?
3. Как из элементов И-НЕ получить функцию ИЛИ?

Практическая работа №2

«Асинхронный RS-триггер» (2 часа)

Цель: собрать и исследовать асинхронный RS-триггер на элементах 2И-НЕ.

Оборудование: микросхема 74НС00 (4 элемента 2И-НЕ), переключатели, светодиоды, резисторы.

Задание:

1. Собрать схему RS-триггера на двух элементах И-НЕ (перекрёстные связи).
2. Подключить входы R и S к кнопкам (с подтяжкой к +5В через 10 кОм, нажатие – GND).
3. Выходы Q и $\neg Q$ – к светодиодам (анод через резистор 330 Ом к выходу, катод на GND).
4. Заполнить таблицу переходов для всех комбинаций R,S (R=0,S=0; 0,1; 1,0; 1,1).
5. Зафиксировать запрещённое состояние (R=1,S=1) и объяснить, почему оно недопустимо.

Порядок выполнения:

- Собрать схему. При включении питания триггер может установиться в произвольное состояние – нажать S=0, чтобы установить Q=1.
- Поочерёдно нажимать кнопки и наблюдать свечение светодиодов.
- Для комбинации R=1,S=1 оба выхода станут 0 (при активном низком уровне) – это нестабильное состояние.

Форма отчёта:

Схема соединений, таблица истинности, временные диаграммы (нарисовать для последовательности S=0, R=0, S=1, R=1). Пояснить, как триггер хранит бит.

Контрольные вопросы:

1. Почему в RS-триггере на И-НЕ активным уровнем является 0?
2. Как изменится работа, если использовать элементы ИЛИ-НЕ?
3. Что такое «состояние гонок» (race condition)?

Практическая работа №3

«Синхронный RS-триггер» (2 часа)

Цель: добавить синхровход С и исследовать работу синхронного RS-триггера.

Оборудование: 74НС00 (И-НЕ), генератор тактовых импульсов (или кнопка с антидребезгом), осциллограф.

Задание:

1. Построить синхронный RS-триггер: добавить к асинхронному два дополнительных И-НЕ на входах R и S, управляемых тактовым сигналом С.
2. Подать на С прямоугольные импульсы (1 Гц, 50% скважность) от генератора или от кнопки (ручной такт).

3. Исследовать, при каких значениях C (высокий/низкий уровень) изменение R и S влияет на выход.

4. Построить временные диаграммы для случаев: $C=1, R=1, S=0$; $C=1, R=0, S=1$; $C=1, R=1, S=1$; $C=0$, изменение R, S не должно менять Q .

Форма отчёта:

Принципиальная схема, таблица истинности (с учётом C), осциллограммы входов и выхода. Вывод о роли синхровхода.

Контрольные вопросы:

1. Чем отличается синхронный триггер от асинхронного?
2. Как реализовать триггер, чувствительный к фронту, а не к уровню?
3. Что такое «защёлка» (latch) и где она применяется?

Практическая работа №4

«Синхронный D-триггер» (2 часа)

Цель: изучить работу D-триггера (схема на основе RS-триггера).

Оборудование: 74НС00, генератор, светодиоды.

Задание:

1. Собрать D-триггер из RS-синхронного, добавив инвертор между R и S ($S = \neg D, R = D$).
2. Подать на тактовый вход C импульсы 1 Гц, на D – переключатель.
3. Убедиться, что при $C=1$ выход Q повторяет D , а при $C=0$ – сохраняет предыдущее состояние.
4. Модифицировать схему для работы по положительному фронту (добавить два последовательных инвертора на линии C или использовать динамический вход).

Порядок выполнения:

- Собрать схему. Для инвертора использовать один элемент И-НЕ с замкнутыми входами.
- Записать таблицу переходов для $D=0$ и $D=1$ при $C=0$ и $C=1$.
- Нарисовать временную диаграмму.

Форма отчёта:

Схема, таблица, диаграмма. Указать, как D-триггер используется для построения регистров.

Контрольные вопросы:

1. Почему D-триггер называют «триггером задержки»?
2. Как из D-триггера получить T-триггер?
3. Где применяется D-триггер в реальной схемотехнике?

Практическая работа №5

«Счётный Т-триггер» (2 часа)

Цель: реализовать Т-триггер на основе D-триггера (или JK) и исследовать режим деления частоты.

Оборудование: D-триггер (микросхема 74НС74) или JK (74НС76), генератор 10 кГц, осциллограф.

Задание:

1. Соединить вход D с инверсным выходом $\neg Q$ (или для JK: $J=K=1$).
2. Подать на тактовый вход T (или C) прямоугольные импульсы частотой $f_{вх}$.
3. Наблюдать на выходе Q сигнал частотой $f_{вых} = f_{вх} / 2$ (деление на 2).
4. Соединить последовательно два Т-триггера – получить деление на 4.
5. Зарисовать осциллограммы входного и выходного сигналов.

Форма отчёта:

Схема соединений, временные диаграммы, расчёт частоты. Вывод о применении Т-триггера в счётчиках.

Контрольные вопросы:

1. Почему Т-триггер называют «счётным»?
2. Как получить деление на нечётный коэффициент?
3. Какое максимальное быстродействие Т-триггера на 74НС74?

Практическая работа №6

«JK-триггер» (2 часа)

Цель: исследовать универсальный JK-триггер во всех режимах.

Оборудование: микросхема 74НС76 (два JK-триггера), генератор, переключатели, осциллограф.

Задание:

1. Исследовать таблицу переходов JK-триггера $(J,K) = (0,0), (0,1), (1,0), (1,1)$ при тактовом сигнале.
2. Убедиться, что режим $J=1, K=1$ работает как Т-триггер (переключение на каждом такте).
3. Проверить, что запрещённой комбинации нет.
4. Собрать схему с обратной связью (например, JK-триггер как делитель на 2).

Порядок выполнения:

- Подать тактовые импульсы 1 кГц.
- Зафиксировать выходы осциллографом для каждой комбинации J,K.
- Обратить внимание на тактирование по спаду (у 74НС76) или по фронту.

Форма отчёта:

Полная таблица переходов, осциллограммы для режима счёта, вывод о преимуществах JK перед RS.

Контрольные вопросы:

1. Как на основе JK-триггера получить D-триггер?
2. Почему JK-триггер используется в промышленных контроллерах?
3. Что такое «мастер-раб» структура?

Практическая работа №7

«Дешифратор» (2 часа)

Цель: собрать и проверить работу дешифратора 3→8 (например, 74НС138).

Оборудование: микросхема 74НС138, светодиоды, переключатели (3 входа, 3 выхода разрешения).

Задание:

1. Подключить три адресных входа (A0-A2) к переключателям.
2. Подключить выходы Y0-Y7 к светодиодам (активный низкий уровень).
3. Зафиксировать на входах разрешения E1=0, E2=0, E3=1 (для 74НС138).
4. Перебирая адрес (000→111), наблюдать, что только один светодиод гаснет (или загорается, если инвертировать).
5. Исследовать влияние входов разрешения: запретить работу (E3=0) – все выходы в 1.

Форма отчёта:

Таблица соответствия адреса и активного выхода, схема подключения. Объяснить, где применяется дешифратор (выбор кристалла памяти, управление семисегментным индикатором).

Контрольные вопросы:

1. Чем отличается дешифратор от демультиплексора?
2. Как построить дешифратор 4→16 на двух микросхемах 3→8?
3. Что такое «приоритетный дешифратор»?

Практическая работа №8

«Шифратор» (2 часа)

Цель: изучить работу приоритетного шифратора 8→3 (74НС148).

Оборудование: 74НС148, кнопки (8 входов), светодиоды на выходах (A0-A2), индикатор GS (Group Select).

Задание:

1. Подключить 8 входов (I0-I7) к кнопкам (активный низкий уровень).

2. На выходах A2-A1-A0 (инверсные) наблюдать двоичный код активного входа с наивысшим приоритетом (I7 – самый высокий).
3. Убедиться, что при нажатии нескольких кнопок кодируется только одна с высшим приоритетом.
4. Записать таблицу истинности для 5-6 комбинаций.

Форма отчёта:

Таблица, схема. Пояснить, зачем нужен выход GS (указывает, есть ли активный вход). Привести пример использования шифратора в клавиатуре.

Контрольные вопросы:

1. Что такое «приоритет» в шифраторе?
2. Как расширить шифратор до 16 входов?
3. Почему выходы шифратора часто инверсные?

Практическая работа №9

«Мультиплексор» (2 часа)

Цель: собрать мультиплексор $4 \rightarrow 1$ (74НС153) и использовать для коммутации сигналов.

Оборудование: 74НС153, переключатели (2 адресных входа, 4 входа данных), осциллограф (или светодиоды).

Задание:

1. Подать на информационные входы D0-D3 логические уровни (0 или 1) от переключателей.
2. На адресные входы S0,S1 подать комбинации (00,01,10,11).
3. На выходе Y наблюдать значение выбранного входа.
4. Усложнить: на D0 подать меандр 1 кГц, на D1 – 2 кГц, на D2 – постоянный 0, D3 – 1. Переключая адрес, наблюдать на осциллографе, как меняется выходной сигнал.

Форма отчёта:

Таблица коммутации, осциллограммы, вывод о применении мультиплексора (коммутация аналоговых/цифровых сигналов, ПЛИС).

Контрольные вопросы:

1. В чём разница между мультиплексором и демультиплексором?
2. Как на мультиплексоре реализовать любую логическую функцию от 3 переменных?
3. Что такое «аналоговый мультиплексор» (например, 74НС4051)?

Практическая работа №10

«Демультиплексор» (2 часа)

Цель: реализовать демультиплексор $1 \rightarrow 4$ на логических элементах или использовать 74НС139.

Оборудование: 74НС139 (двойной демультиплексор $1 \rightarrow 4$), источник сигнала, светодиоды.

Задание:

1. На вход данных подать логический уровень (1).
2. На адресные входы (A_0, A_1) подать комбинации.
3. Наблюдать, на каком из четырёх выходов появится сигнал (активный низкий уровень).
4. Поменять входной сигнал на 0 – убедиться, что активный выход также 0.
5. Собрать схему, где демультиплексор используется для выбора одного из четырёх светодиодов.

Форма отчёта:

Схема, таблица истинности, пример использования демультиплексора в адресации памяти.

Контрольные вопросы:

1. Можно ли использовать дешифратор как демультиплексор?
2. Чем отличается демультиплексор от коммутатора?
3. Как построить демультиплексор $1 \rightarrow 16$?

Практическая работа №11

«Сумматор» (2 часа)

Цель: собрать и проверить работу полусумматора и полного одноразрядного сумматора.

Оборудование: микросхемы 74НС86 (XOR), 74НС08 (AND), 74НС32 (OR), светодиоды.

Задание:

1. Собрать полусумматор (два входа A, B ; выходы: сумма $S = A \oplus B$, перенос $C = A \& B$).
2. Проверить таблицу истинности.
3. Собрать полный сумматор на двух полусумматорах и ИЛИ (входы A, B, C_{in} ; выходы S, C_{out}).
4. Заполнить таблицу для всех 8 комбинаций.
5. Соединить два полных сумматора – получить 2-разрядный сумматор.

Форма отчёта:

Схемы, таблицы истинности, временные диаграммы для сложения (например, $A=1, B=1, Cin=0$). Вывод о роли сумматора в АЛУ.

Контрольные вопросы:

1. Как на сумматоре реализовать вычитание?
2. Что такое «сумматор с ускоренным переносом»?
3. Какова максимальная частота работы 4-разрядного сумматора?

Практическая работа №12

«Суммирующий счётчик» (2 часа)

Цель: построить 3-разрядный двоичный суммирующий счётчик на JK-триггерах.

Оборудование: микросхемы 74НС76 (два JK-триггера, всего 4 – но можно использовать два корпуса), генератор, светодиоды.

Задание:

1. Собрать схему трёхразрядного асинхронного счётчика (каждый триггер – в режиме Т, такт на вход первого, выход Q_0 – на тактовый вход второго и т.д.).
2. Подать тактовые импульсы 1 Гц (визуально) или 1 кГц (осциллограф).
3. Наблюдать состояние $Q_2Q_1Q_0$ в двоичном коде ($0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow 7 \rightarrow 0$).
4. Построить временные диаграммы для 8 тактов.

Форма отчёта:

Схема, диаграммы, таблица состояний. Объяснить, как изменить направление счёта (вычитающий счётчик).

Контрольные вопросы:

1. Почему такой счётчик называется «асинхронным»?
2. Как сделать сброс счётчика в 0?
3. Где применяются счётчики в реальных устройствах (частотомеры, таймеры)?

Практическая работа №13

«Вычитающий счётчик» (2 часа)

Цель: построить 3-разрядный вычитающий счётчик.

Оборудование: те же JK-триггеры, инверторы.

Задание:

1. Модифицировать схему суммирующего счётчика так, чтобы счёт шёл вниз ($Q_2Q_1Q_0$: $7 \rightarrow 6 \rightarrow \dots \rightarrow 0 \rightarrow 7$).
2. Для этого тактовые сигналы подавать не с прямых выходов, а с инверсных ($\neg Q$).

3. Собрать схему, убедиться в работе.
4. Объединить суммирующий и вычитающий режимы с помощью переключателя (реверсивный счётчик).

Форма отчёта:

Схема, таблица переходов, диаграммы. Вывод о возможности построения реверсивного счётчика.

Контрольные вопросы:

1. Как получить счётчик с произвольным модулем (не 8)?
2. В чём отличие синхронного счётчика от асинхронного?
3. Какую максимальную частоту может обеспечить такой счётчик?

3. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К ПРАКТИЧЕСКИМ РАБОТАМ (РАЗДЕЛ 2. МИКРОКОНТРОЛЛЕРНЫЕ СИСТЕМЫ)

Практическая работа №1

«Мигание светодиодом» (4 часа)

Цель: освоить среду Arduino IDE, написать первую программу для управления встроенным светодиодом.

Оборудование: плата Arduino Uno, светодиод (если встроенного мало), резистор 220 Ом.

Задание:

1. Подключить плату, установить драйверы.
2. Написать программу, которая включает и выключает светодиод на pin 13 с периодом 1 секунда.
3. Изменить скважность (время вкл/выкл) на 250/750 мс.
4. Подключить внешний светодиод (через резистор) к pin 9 и повторить программу.
5. Сделать так, чтобы светодиод мигал 5 раз, затем пауза 2 секунды, и снова 5 раз (цикл).

Методические указания:

Использовать `pinMode(LED_BUILTIN, OUTPUT)`, `digitalWrite()`, `delay()`. Объяснить, почему `delay()` – плохо для сложных проектов.

Форма отчёта:

Листинг с комментариями, схема подключения, фотография работающего устройства.

Контрольные вопросы:

1. Зачем нужен резистор последовательно со светодиодом?
2. Что произойдёт, если подать 5 В на вывод, настроенный как INPUT?
3. Какой максимальный ток можно получить с вывода Arduino?

Практическая работа №2

«Бегущие огни на светодиодах» (4 часа)

Цель: научиться управлять несколькими светодиодами, использовать массивы и циклы.

Оборудование: 8 светодиодов, резисторы 220 Ом (8 шт.), макетная плата.

Задание:

1. Подключить 8 светодиодов к выводам 2–9 через резисторы.
2. Написать программу, которая последовательно зажигает и гасит каждый светодиод (эффект бегущей строки) с интервалом 100 мс.

3. Реализовать два режима: движение влево, движение вправо, переключение по нажатию кнопки.

4. Добавить режим «накопление» (сначала горит 1, потом 1 и 2, потом 1,2,3... затем все гаснут).

Методические указания:

Использовать массив пинов `int pins[] = {2,3,4,5,6,7,8,9};`. Для задержки – `delay()` пока допустимо.

Форма отчёта:

Листинг, схема, описание работы кнопки (с антидребезгом – простейший через `delay(50)`).

Контрольные вопросы:

1. Почему при большом количестве светодиодов нельзя питать их прямо от выводов МК (суммарный ток)?
2. Как можно реализовать бегущие огни без `delay()`?
3. Что такое «сдвиговый регистр» (74HC595) и зачем он нужен?

Практическая работа №3

«Подсчёт нажатий кнопки» (4 часа)

Цель: научиться обрабатывать кнопку с подавлениемдребезга и выводить результат на последовательный порт.

Оборудование: кнопка, резистор подтяжки 10 кОм, Arduino, компьютер.

Задание:

1. Подключить кнопку между pin 2 и GND, подтяжку включить внутреннюю (`pinMode(2, INPUT_PULLUP)`).
2. Написать программу, которая считает количество нажатий (счётчик увеличивается на 1 при каждом отпускании кнопки).
3. Выводить значение счётчика в Serial Monitor.
4. Реализовать программное подавлениедребезга (задержка 30-50 мс после обнаружения изменения).
5. Добавить сброс счётчика при нажатии другой кнопки или через определённое время.

Методические указания:

Использовать `Serial.begin(9600)`, `Serial.println(counter)`. Для антидребезга – запоминать время последнего переключения (`millis()`).

Форма отчёта:

Листинг, схема, скриншот монитора порта. Пояснить, почему без антидребезга счётчик «удваивает» нажатия.

Контрольные вопросы:

1. Чем отличается `INPUT` от `INPUT_PULLUP`?
2. Какие ещё способы подавления дребезга существуют (аппаратные)?
3. Как реализовать длительное и короткое нажатие?

Практическая работа №4

«Создание временного интервала при помощи таймера» (4 часа)

Цель: освоить неблокирующие задержки с использованием `millis()` и аппаратных таймеров.

Оборудование: Arduino, светодиод, кнопка.

Задание:

1. Написать программу, в которой светодиод мигает с периодом 1 секунда (500/500), но при этом программа опрашивает кнопку и немедленно реагирует на нажатие (например, включает другой светодиод).
2. Запретить `delay()`.
3. Реализовать три режима мигания, переключаемых кнопкой: 100 мс, 500 мс, 1000 мс.
4. (Дополнительно) Использовать аппаратный таймер `Timer1` (библиотека `TimerOne`) для генерации прерываний.

Методические указания:

Использовать переменные `previousMillis`, `currentMillis`, `interval`. В `loop()` проверять, не пора ли переключить светодиод.

Форма отчёта:

Листинг с пояснениями, блок-схема конечного автомата. Сравнение с `delay()`.

Контрольные вопросы:

1. Что произойдёт с `millis()` через 50 дней работы?
2. Как реализовать несколько независимых таймеров (для разных задач)?
3. В чём преимущество аппаратного таймера перед `millis()`?

Практическая работа №5

«Формирование ШИМ» (4 часа)

Цель: научиться использовать ШИМ-выходы для управления яркостью светодиода и скоростью двигателя.

Оборудование: Arduino, светодиод (или RGB), резистор, потенциометр (для задания яркости), двигатель постоянного тока с драйвером (например, L298N – опционально).

Задание:

1. Подключить светодиод к ШИМ-выводу (например, pin 9).
2. Написать программу, которая плавно увеличивает и уменьшает яркость (эффект «дыхания»). Использовать `analogWrite()`.
3. Подключить потенциометр к аналоговому входу A0. Считывать значение (0–1023) и пересчитывать в яркость (0–255). Управлять яркостью светодиода поворотом ручки.
4. (Для сильных) Подключить двигатель через драйвер и регулировать скорость ШИМ.

Методические указания:

Частота ШИМ на Arduino Uno – ~490 Гц (на pin 5,6 – ~980 Гц). Для визуального изменения яркости достаточно.

Форма отчёта:

Схема, листинг, фотография меняющейся яркости. Осциллограмма ШИМ (если есть осциллограф).

Контрольные вопросы:

1. Что такое скважность ШИМ?
2. Как изменить частоту ШИМ (таймеры)?
3. Для чего используется ШИМ в управлении серводвигателем?

Практическая работа №6

«Система прерываний» (4 часа)

Цель: изучить внешние прерывания для обработки кнопки без опроса в цикле.

Оборудование: Arduino, кнопка, светодиод.

Задание:

1. Подключить кнопку к выводу 2 (внешнее прерывание INT0).
2. Написать программу, в которой светодиод мигает с периодом 1 секунда (используя `millis()`), а при нажатии кнопки (прерывание) счётчик нажатий увеличивается и выводится в Serial.
3. В обработчике прерывания выполнять минимум действий (только установка флага).
4. Реализовать «тихий режим»: двойное нажатие кнопки отключает мигание светодиода, ещё одно двойное – включает.

Методические указания:

Использовать `attachInterrupt(digitalPinToInterrupt(2), handler, FALLING)`. Флаг – `volatile bool`. В основном цикле проверять флаг и выполнять действия.

Форма отчёта:

Листинг, пояснение, почему в обработчике нельзя использовать `delay()` или `Serial.print()` (долго).

Контрольные вопросы:

1. Какие виды прерываний бывают (по уровню, по фронту)?
2. Что такое «реентерабельность» и почему важна в обработчике?
3. Как отключить прерывания на время критической секции?

4. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К УЧЕБНОЙ ПРАКТИКЕ (72 часа)

Учебная практика проводится после освоения раздела 2 и включает 9 видов работ.

Каждый вид – примерно 8 часов.

Вид работы 1. Использование АЦП микроконтроллера

Задание:

Подключить аналоговый датчик (потенциометр, фоторезистор, датчик Холла). Написать программу, которая каждые 200 мс считывает значение АЦП (10 бит), преобразует в напряжение и выводит в Serial. Построить график зависимости (можно в Excel).

Реализовать автоматическую калибровку (запоминание min/max).

Результат: отчёт с кодом, схемой, графиком.

Вид работы 2. Взаимодействие с встроенной памятью EEPROM

Задание:

Сохранять в EEPROM значение счётчика нажатий кнопки при выключении питания (имитация – при нажатии другой кнопки «сохранить»). При включении – считывать последнее сохранённое значение. Использовать библиотеку EEPROM.

Продемонстрировать, что после перезагрузки счётчик не обнуляется.

Вид работы 3. Взаимодействие со светодиодной матрицей (например, 8×8 MAX7219)

Задание:

Вывести на матрицу бегущую строку с текстом «SAKHGU». Реализовать сдвиг влево/вправо по кнопкам. Использовать библиотеку LedControl.

Вид работы 4. Взаимодействие с ЖКИ (символьный дисплей 16×2, I2C или параллельный)

Задание:

Выводить на дисплей температуру с датчика DS18B20 (см. вид 5) и время работы системы в секундах. Организовать меню из двух экранов (переключение по кнопке).

Вид работы 5. Работа с цифровым температурным датчиком (DS18B20, 1-Wire)

Задание:

Подключить один или два датчика на одну шину. Считывать температуру с точностью 0.5°C. Выводить на ЖКИ и в Serial. Реализовать сигнализацию при превышении порога (светодиод или зуммер).

Вид работы 6. Работа с двигателем постоянного тока (через драйвер L298N или транзистор)

Задание:

Управлять скоростью вращения двигателя с помощью ШИМ (от потенциометра).

Реализовать реверс (два направления) с помощью H-моста. Плавный пуск и торможение.

Вид работы 7. Работа с серводвигателем (SG90 или MG995)**Задание:**

Подключить сервопривод к ШИМ-выводу (библиотека Servo). Поворачивать вал в зависимости от значения потенциометра (0° – 180°). Организовать автоматическое качание (от 0 до 180 и обратно с шагом 1°).

Вид работы 8. Работа с шаговым двигателем (28BYJ-48 с драйвером ULN2003)**Задание:**

Реализовать вращение шагового двигателя в обе стороны с заданным числом шагов (например, 2048 полных шагов – один оборот). Управление кнопками: «влево», «вправо», «стоп». Регулировка скорости (задержка между шагами).

Вид работы 9. Работа с модулем передачи информации (например, Bluetooth HC-05 или LoRa, Wi-Fi ESP8266)**Задание:**

Связать два модуля (или МК с телефоном). Передавать данные датчика температуры с одного устройства на другое, отображать на ЖКИ или в Serial. Реализовать управление светодиодом удалённой командой (например, "LED_ON").

Общие требования к отчёту по учебной практике:

- Краткое описание цели и оборудования.
- Принципиальная схема подключения.
- Листинг программы с комментариями.
- Фото/видео работающего устройства.
- Выводы (что освоил, какие были трудности).

5. КРИТЕРИИ ОЦЕНКИ ПРАКТИЧЕСКИХ РАБОТ

Оценка	Требования
5 (отлично)	Работа выполнена в полном объёме, схема собрана правильно, программа загружена и функционирует без ошибок. Отчёт оформлен аккуратно, содержит все разделы, выводы. Студент свободно отвечает на контрольные вопросы.
4 (хорошо)	Работа выполнена, но есть 1-2 недочёта (например, не полностью реализовано дополнительное задание, отчёт без выводов). На вопросы отвечает с небольшими затруднениями.
3 (удовлетворительно)	Работа выполнена частично (не менее 60%), но основные функции работают. Отчёт небрежный, нет схемы. При защите показывает минимальное понимание.
2 (неудовлетворительно)	Работа не выполнена или не работает основная функция. Отсутствует отчёт. Студент не может объяснить схему или код.

6. ЛИТЕРАТУРА И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Основные источники (из РП):

1. Степина В.В. Архитектура ЭВМ и вычислительные системы. – М.: ИНФРА-М, 2021.
2. Миленина С.А. Электроника и схемотехника. – М.: Юрайт, 2021. (ЭБС)
3. Сажнев А.М. Микропроцессорные системы. – М.: Юрайт, 2021.

Программное обеспечение:

- Arduino IDE (бесплатно)
- Logisim / Proteus (для симуляции схем)
- Atmel Studio / Microchip Studio (для AVR C)
- Serial Monitor, Tinkercad Circuits (онлайн)

Интернет-ресурсы:

- Arduino Reference (arduino.cc/reference)
- Datasheets на микросхемы (alldatasheet.com)
- Видеоуроки по работе с датчиками и двигателями.