

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Умаров Марат Файзуллаевич
Должность: Директор
Дата подписания: 16.02.2026 11:50:12
Уникальный программный ключ:
48505f11ec15acaa386f5219d3113d7276efda78

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ


Федеральное государственное автономное образовательное учреждение высшего
образования

"Казанский (Приволжский) федеральный университет"

Елабужский институт (филиал) КФУ



УТВЕРЖДАЮ

Директор
Елабужского института КФУ
 Е.Е. Мерзон.
" 8 " 06 2023 г.

Программа дисциплины (модуля)

Программирование и робототехника

Направление подготовки: 44.03.05 - Педагогическое образование (с двумя профилями подготовки)

Профиль подготовки: Математика и информатика

Квалификация выпускника: бакалавр

Форма обучения: очная

Язык обучения: русский

Год начала обучения по образовательной программе: 2023

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО
2. Место дисциплины (модуля) в структуре ОПОП ВО
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся
4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий
 - 4.1. Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)
 - 4.2. Содержание дисциплины (модуля)
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)
6. Фонд оценочных средств по дисциплине (модулю)
7. Перечень литературы, необходимой для освоения дисциплины (модуля)
8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)
9. Методические указания для обучающихся по освоению дисциплины (модуля)
10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)
11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)
12. Средства адаптации преподавания дисциплины (модуля) к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья
13. Приложение №1. Фонд оценочных средств
14. Приложение №2. Перечень литературы, необходимой для освоения дисциплины (модуля)
15. Приложение №3. Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Программу дисциплины разработал(а)(и) доцент, к.н. (доцент) Минкин А.В. (Кафедра математики и прикладной информатики)

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО

Обучающийся, освоивший дисциплину (модуль), должен обладать следующими компетенциями:

Шифр компетенции	Расшифровка приобретаемой компетенции
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК-1.1	Знать принципы поиска информации, критического анализа и синтеза информации, методики системного подхода для решения поставленных задач
УК-1.2	Уметь осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК-1.3	Владеть навыками поиска, критического анализа и синтеза информации; способностью применять системный подход для решения поставленных задач
УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
УК-2.1	Знать требования к определению задач в рамках поставленной цели; способы решения задач с учетом действующих правовых норм, имеющихся ресурсов и ограничений
УК-2.2	Уметь определять круг задач в рамках поставленной цели, выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений.
УК-2.3	Владеть навыками определения круга задач в рамках поставленной цели, выбора оптимальных способов их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
ОПК-2	Способен участвовать в разработке основных и дополнительных образовательных программ, разрабатывать отдельные их компоненты (в том числе с использованием информационно-коммуникационных технологий)
ОПК-2.3	Владеть навыками разработки в составе команды основных и дополнительных образовательных программ, их отдельных компонентов (в том числе с использованием информационно-коммуникационных технологий)

Обучающийся, освоивший дисциплину (модуль):

Должен знать:

эффективные технологии организации контроля и оценки формирования результатов образования обучающихся, выявления и коррекции трудностей в обучении программированию и робототехнике; требования к определению круга задач в рамках поставленной цели; способы эффективного решения задач программированию и робототехнике с учетом действующих правовых норм, имеющихся ресурсов и ограничений.

Должен уметь:

выбирать и применять эффективные технологии организации контроля и оценки формирования результатов образования обучающихся, выявления и коррекции трудностей в обучении программированию и робототехнике; определять круг задач программированию и робототехнике задач в рамках поставленной цели, выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений.

Должен владеть:

эффективными технологиями осуществления контроля и оценки формирования результатов образования обучающихся, выявления и коррекции трудностей в обучении, возникающих при решении задач программирования;

навыками определения круга задач по программированию и робототехнике в рамках поставленной цели, выбора оптимальных способов их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений;

навыками разработки в составе команды основных и дополнительных образовательных программ (в том числе с использованием программирования и робототехники) с учетом особенностей конкретной образовательной среды и индивидуальных потребностей обучающихся

2. Место дисциплины (модуля) в структуре ОПОП ВО

Данная дисциплина (модуль) включена в раздел "Б1.О.08.05 Дисциплины (модули)" основной профессиональной образовательной программы 44.03.05 "Педагогическое образование (с двумя профилями подготовки) (Математика и информатика)" и относится к обязательным дисциплинам. Осваивается на 3 курсе в 5 семестре и на 4 курсе в 8 семестре.

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 9 зачетных(ые) единиц(ы) на 324 часа(ов).

Контактная работа - 182 часа(ов), в том числе лекции - 62 часа(ов), практические занятия - 0 часа(ов), лабораторные работы - 120 часа(ов), контроль самостоятельной работы - 0 часа(ов).

Самостоятельная работа - 106 часа(ов).

Контроль (зачёт / экзамен) - 36 часа(ов).

Форма промежуточного контроля дисциплины: зачет в 5 семестре; экзамен в 8 семестре.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1 Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)

N	Разделы дисциплины / модуля	Семестр	Виды и часы контактной работы, их трудоемкость (в часах)			Самостоятельная работа
			Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. Программирование Arduino, введение. Структура программы, константы. Цифровой ввод/вывод. Аналоговый ввод/вывод.	5	8	0	12	22
2.	Тема 2. Дополнительные функции ввода/вывода. Работа со временем. Математические функции. Псевдослучайные числа	5	8	0	14	22
3.	Тема 3. Последовательная передача данных. Прерывания. EEPROM. Blink без delay.	5	10	0	14	22
4.	Тема 4. Создание библиотеки. Ethernet, Servo, Firmata library, работа с библиотеками.	5	10	0	14	24
5.	Тема 5. Цифровой ввод, кнопка. Аналоговый вывод, Fading. Аналоговый ввод, потенциометр и осциллограф	8	6	0	14	2
6.	Тема 6. Генерация звука, пьезоизлучатель. Фоторезистор. Сенсор на светодиоде. Общение с Arduino - программирование работы с COM-портом. Подключаем к Arduino мышку PS/2	8	6	0	14	4
7.	Тема 7. Аналоговый датчик температуры, LM335. Протокол 1-Wire и iButton. Arduino и эмулятор iButton. Arduino и температурный 1-Wire датчик DS18S20.	8	6	0	14	4
8.	Тема 8. Arduino и драйвер двигателей L293D (Простой мотор-шилд). Arduino и сервомашинка. LCD-дисплей на базе HD44780 и Arduino.	8	4	0	12	4
9.	Тема 9. ИК-датчик препятствий для Arduino на базе фототранзистора	8	4	0	12	2
	Итого: 324 ч. (из них 36 ч. контроль)		62	0	120	108

4.2 Содержание дисциплины (модуля)

Тема 1. Программирование Arduino, введение. Структура программы, константы. Цифровой ввод/вывод. Аналоговый ввод/вывод.

Знакомство и начало работы с Arduino Uno: о том что это и зачем она нужна; о подключении к компьютеру и установке программной среды на ОС Windows;

Знакомство со средой программирования, структурой программы, используемые типы переменных и написание первой программы для Arduino.

Подключении кнопок к Arduino Uno; о том зачем нужен breadboard. Объясняется как применяется широтно-импульсная модуляция (также известная как ШИМ или PWM). Рассматривается возможность создавать собственные функции в среде программирования Arduino на примере функции стабилизации неустойчивого сигнала.

Использование аналоговых входов Arduino. С помощью схемы делителя напряжения и фоторезистора собираем фотосенсор. На основе него делаем прототип устройства резервного освещения. И возможности применения инфракрасного дальномера.

Тема 2. Дополнительные функции ввода/вывода. Работа со временем. Математические функции. Псевдослучайные числа

Рассматриваются базовые понятия схемотехники. О токе и напряжении, о том как применяются резисторы и как строятся делители напряжения. О стягивающих, подтягивающих и токоограничивающих резисторах. О законе Ома. О сенсорах и регуляторах напряжения. О подстроечных резисторах и аналоговых входах. И о том, как всё это соотносится с Arduino.

Рассматриваются дополнительные специфичные функции Arduino для работы со временем `unsigned long millis(void)`; Вызов: `time = millis()`;

Рассматриваются специфичные математические функции Arduino: `min(x, y)`, `max(x, y)`, `abs(x)`, `constrain(x, a, b)`, `map(value, fromLow, fromHigh, toLow, toHigh)`, `pow(base, exponent)`, `sq(x)`, `sqrt(x)`, `sin(rad)`, `cos(rad)`, `tan(rad)`.

Тема 3. Последовательная передача данных. Прерывания. EEPROM. Blink без delay.

Рассматривается Arduino/Arduino встроенный контроллер для последовательной передачи данных, который может использоваться как для связи между Arduino/Arduino устройствами, так и для связи с компьютером. На компьютере соответствующее соединение представлено либо обычным COM-портом (в случае Arduino Single-Sided Serial Board), либо USB COM-портом, который появляется в системе после установки необходимого драйвера.

Знакомство с прерываниями. Прерывание (англ. interrupt) ? сигнал, сообщающий процессору о наступлении какого-либо события. При этом выполнение текущей последовательности команд приостанавливается, и управление передаётся обработчику прерывания, который выполняет работу по обработке события и возвращает управление в прерванный код.

Знакомство с EEPROM. EEPROM, (Electrically Erasable Programmable Read-Only Memory) электрически стираемое перепрограммируемое ПЗУ, ЭСППЗУ). Память такого типа может стираться и заполняться данными несколько десятков тысяч раз. Используется в твердотельных накопителях. Одной из разновидностей EEPROM является флеш-память (Flash Memory).

Пример использования `millis()` для, одновременного, выполнения нескольких задач.

Тема 4. Создание библиотеки. Ethernet, Servo, Firmata library, работа с библиотеками.

На примере библиотеки для работы с PS/2-устройствами оценивается работа через библиотеку. Библиотека, это дополнительный класс, который инкапсулирует в себе функции для работы с устройством, т.о. для создания библиотеки нужно создать один .h файл в котором будет описываться класс/функции/ константы и один .cpp, в котором все эти функции реализуются.

Ethernet-шилд, это дополнительная плата, которая подключается к ардуино и позволяет этой замечательной плате посылать и получать данные через локальную сеть и даже работать Интернетом. Работа с протоколами TCP/IP в шилде реализована аппаратно на микросхеме W5100.

Servo library - библиотека для работы с сервомашинками. Сервомашинка - это мотор-редуктор, с обратной связью, благодаря которой можно повернуть выходной вал на строго определённый угол. Стандартные рулевые машинки, вертятся от 0 до 180 градусов.

Библиотека Firmata реализует протокол Firmata, что позволяет простым образом общаться с программами на компьютере. Данная библиотека входит в состав Arduino IDE. Удобна тем, что при необходимости не нужно изобретать велосипед и придумывать свой протокол, а уже использовать этот готовый протокол.

Тема 5. Цифровой ввод, кнопка. Аналоговый вывод, Fading. Аналоговый ввод, потенциометр и осциллограф

Рассматривается скетч Fading, как пример работы с функцией `analogWrite`, аналоговым выводом сигнала. Демонстрируется, что значение, которое можно передавать в `analogWrite` должно быть между 0 и 255, что

соответствует напряжению от 0V до 5V на выходе порта. Постепенно увеличивая/уменьшая переданное в analogWrite значение можно добиться плавного роста/падения напряжения на выходе. Подключив к одному из PWM-портов ардуины (3, 5, 6, 9, 10, 11, а на платах на базе ATmega8 только, 9, 10, 11) светодиод будет плавно загораться и затухать (fading).

Рассматривается аналоговый ввод, с использованием переменного резистора, подключённый к аналоговому входу. Вводится функция, analogRead(), которая считывает значение с аналогового порта. Демонстрируется простой осциллограф. Аналоговый ввод, осциллограф.

Тема 6. Генерация звука, пьезоизлучатель. Фоторезистор. Сенсор на светодиоде. Общение с Arduino - программирование работы с СОМ-портом. Подключаем к Arduino мышку PS/2

Пьезокерамические излучатели (пьезоизлучатели), электроакустические устройства воспроизведения звука, использующие пьезоэлектрический эффект (эффект возникновения поляризации диэлектрика под действием механических напряжений (прямой пьезоэлектрический эффект). Существует и обратный пьезоэлектрический эффект, возникновение механических деформаций под действием электрического поля.

Фоторезистор, полупроводниковый прибор, изменяющий величину своего сопротивления при облучении светом.

Рассмотрено нестандартное использование светодиода.

Рассматривается консольная POSIX C, программа, программа может принимать и передавать данные на плату Arduino.

Тема 7. Аналоговый датчик температуры, LM335. Протокол 1-Wire и iButton. Arduino и эмулятор iButton. Arduino и температурный 1-Wire датчик DS18S20.

Рассмотрен вариант использования терморезистора. Рассмотрено, как работать с аналоговыми датчиками температуры, на примере LM335. Знакомство с 1-Wire и работа с устройством, использующим этот протокол на примере подключения к Arduino температурного 1-Wire датчика DS18S20. Рассматривается Arduino и эмулятор iButton.

Тема 8. Arduino и драйвер двигателей L293D (Простой мотор-шилд). Arduino и сервомашинка. LCD-дисплей на базе HD44780 и Arduino.

Используется популярный драйвер для управления моторчиками, L293D. L293D содержит два драйвера для управления электродвигателями небольшой мощности. Имеет две пары входов для управляющих сигналов и две пары выходов для подключения электромоторов. Кроме того, у L293D есть два входа для включения каждого из драйверов. Эти входы используются для управления скоростью вращения электромоторов с помощью ШИМ (PWM).

Рассматривается мотор-редуктор, способный поворачивать выходной вал строго в заданное положение (на угол) и удерживать его там, вопреки сопротивлениям и возмущениям недружелюбной среды.

Рассматривается LCD-дисплеи на базе контроллера HD44780 на примере WH1602B-YYK-CTK

Тема 9. ИК-датчик препятствий для Arduino на базе фототранзистора

Рассматривается аналоговый сенсор на фототранзисторе. Так как датчик аналоговый, то его выход должен подключаться к аналоговым портам контроллера Arduino (на вход АЦП микроконтроллера). По величине аналогового сигнала мы сможем примерно оценивать расстояние до препятствия (разумеется, абсолютных величин мы получить не сможем, так как уровень сигнала будет меняться в зависимости от объекта). Простейшая схема, это пара из ИК-светодиода и фототранзистора.

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Самостоятельная работа обучающихся выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Самостоятельная работа подразделяется на самостоятельную работу на аудиторных занятиях и на внеаудиторную самостоятельную работу. Самостоятельная работа обучающихся включает как полностью самостоятельное освоение отдельных тем (разделов) дисциплины, так и проработку тем (разделов), осваиваемых во время аудиторной работы. Во время самостоятельной работы обучающиеся читают и конспектируют учебную, научную и справочную литературу, выполняют задания, направленные на закрепление знаний и отработку умений и навыков, готовятся к текущему и промежуточному контролю по дисциплине.

Организация самостоятельной работы обучающихся регламентируется нормативными документами, учебно-методической литературой и электронными образовательными ресурсами, включая:

Порядок организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры (утвержденный приказом Министерства науки и высшего образования Российской Федерации от 6 апреля 2021 года № 245)

Устав федерального государственного автономного образовательного учреждения "Казанский (Приволжский) федеральный университет"

Правила внутреннего распорядка федерального государственного автономного образовательного учреждения высшего профессионального образования "Казанский (Приволжский) федеральный университет"

Локальные нормативные акты Казанского (Приволжского) федерального университета

6. Фонд оценочных средств по дисциплине (модулю)

Фонд оценочных средств по дисциплине (модулю) включает оценочные материалы, направленные на проверку освоения компетенций, в том числе знаний, умений и навыков. Фонд оценочных средств включает оценочные средства текущего контроля и оценочные средства промежуточной аттестации.

В фонде оценочных средств содержится следующая информация:

- соответствие компетенций планируемым результатам обучения по дисциплине (модулю);
- критерии оценивания сформированности компетенций;
- механизм формирования оценки по дисциплине (модулю);
- описание порядка применения и процедуры оценивания для каждого оценочного средства;
- критерии оценивания для каждого оценочного средства;
- содержание оценочных средств, включая требования, предъявляемые к действиям обучающихся, демонстрируемым результатам, задания различных типов.

Фонд оценочных средств по дисциплине находится в Приложении 1 к программе дисциплины (модулю).

7. Перечень литературы, необходимой для освоения дисциплины (модуля)

Освоение дисциплины (модуля) предполагает изучение учебной литературы. Литература может быть доступна обучающимся в одном из двух вариантов (либо в обоих из них):

- в электронном виде - через электронные библиотечные системы на основании заключенных КФУ договоров с правообладателями;
- в печатном виде - в Научной библиотеке Елабужского института КФУ. Обучающиеся получают учебную литературу на абонементе по читательским билетам в соответствии с правилами пользования Научной библиотекой.

Электронные издания доступны дистанционно из любой точки при введении обучающимся своего логина и пароля от личного кабинета в системе "Электронный университет". При использовании печатных изданий библиотечный фонд должен быть укомплектован ими из расчета не менее 0,25 экземпляра на каждого обучающегося из числа лиц, одновременно осваивающих данную дисциплину.

Перечень учебной литературы, необходимой для освоения дисциплины (модуля), находится в Приложении 2 к рабочей программе дисциплины. Он подлежит обновлению при изменении условий договоров КФУ с правообладателями электронных изданий и при изменении комплектования фондов Научной библиотеки Елабужского института КФУ

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Библиотека по робототехнике - <http://roboticslib.ru/books/>

Научно-популярный портал Занимательная робототехника - <http://edurobots.ru/category/uroki/>

Сайт компании «Амперка» - <http://www.amperka.ru>

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Вид работ	Методические рекомендации
лекции	При изучении дисциплины сначала необходимо по каждой теме лекции прочитать рекомендованную литературу и составить краткий конспект основных положений, терминов, сведений, требующих запоминания и являющихся основополагающими в этой теме для освоения последующих тем курса. Для расширения знания по дисциплине рекомендуется использовать Интернет-ресурсы; проводить поиски в различных системах и использовать материалы сайтов, рекомендованных преподавателем.
лабораторные	При подготовке к лабораторным занятиям необходимо заранее изучить

Вид работ	Методические рекомендации
работы	методические рекомендации по его проведению. Обратить внимание на цель работы, на основные вопросы для подготовки к работе, на содержание темы работы. Лабораторное занятие проходит в виде диалога, разбора основных вопросов темы. Также лабораторное занятие может проходить в виде показа презентаций, демонстративного материала (в частности плакатов, слайдов), которые сопровождаются беседой преподавателя со студентами. Студент может сдавать лабораторную работу в виде написания реферата, подготовки слайдов, презентаций и последующей защиты его, либо может написать конспект в тетради, ответив на вопросы по заданной теме. Ответы на вопросы можно сопровождать рисунками, схемами и т.д. с привлечением дополнительной литературы, которую следует указать.
самостоятельная работа	Обучающийся самостоятельно определяет режим своей самостоятельной работы и меру труда, затрачиваемого на овладение знаниями и умениями по дисциплине, выполняет внеаудиторную работу по индивидуальному плану, в зависимости от собственной подготовки, бюджета времени и других условий. Ежедневно обучающийся должен уделять выполнению самостоятельной работы в среднем не менее 3 часов. При выполнении самостоятельной работы обучающийся имеет право обращаться к преподавателю за консультацией с целью уточнения задания, формы контроля выполненного задания.
зачет	Зачет нацелен на комплексную проверку освоения дисциплины. Зачет проводится в устной или письменной форме по билетам, в которых содержатся вопросы (задания) по всем темам курса. Обучающемуся даётся время на подготовку. Оценивается владение материалом, его системное освоение, способность применять нужные знания, навыки и умения при анализе проблемных ситуаций и решении практических заданий.
экзамен	Для контроля усвоения данной дисциплины предусмотрен экзамен, на котором студентам необходимо ответить на вопросы экзаменационных билетов. При ответе на экзамене необходимо: продумать и четко изложить материал; дать определение основных понятий; дать краткое описание явлений; привести примеры. Ответ следует иллюстрировать схемами, рисунками и графиками.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем, представлен в Приложении 3 к рабочей программе дисциплины (модуля).

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория №60 (423600, Республика Татарстан, г. Елабуга, ул. Казанская, д. 89) для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, помещения для самостоятельной работы. Площадь 73,4 кв.м. Комплект мебели (посадочных мест) 29 шт. Комплект мебели (посадочных мест) для преподавателя 1 шт. Компьютерный класс: Компьютеры intel core i5 15 шт. Мониторы ViewSonic 22d 15 шт. Проектор EPSON EB-535W 1 шт. Интерактивная доска IQBoard DVT TN082 1 шт. Трибуна 1 шт. Кондиционер 1 шт. Настенные полки 6 шт. Шкаф двухстворчатый с полками 1 шт. Веб-камера 1 шт. Выход в Интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду. Набор учебно-наглядных пособий: комплект презентаций в электронном формате по преподаваемой дисциплине 3-5 шт.

12. Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья

При необходимости в образовательном процессе применяются следующие методы и технологии, облегчающие восприятие информации обучающимися инвалидами и лицами с ограниченными возможностями здоровья:

- создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;
- создание контента, который можно представить в различных видах без потери данных или структуры,

предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;

- создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников - например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения - аудиально;

- применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной, за счёт альтернативных способов, в том числе виртуальных лабораторий и симуляционных технологий;

- применение дистанционных образовательных технологий для передачи информации, организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных работ, проведения тренингов, организации коллективной работы;

- применение дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;

- увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи:

- продолжительности сдачи зачёта или экзамена, проводимого в письменной форме, - не более чем на 90 минут;

- продолжительности подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, - не более чем на 20 минут;

- продолжительности выступления обучающегося при защите курсовой работы - не более чем на 15 минут.

Программа составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению 44.03.05 "Педагогическое образование (с двумя профилями подготовки)" и профилю подготовки "Математика и информатика".

*Приложение №1
к рабочей программе дисциплины (модуля)
Б1.О.08.05 Программирование и робототехника*

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
"Казанский (Приволжский) федеральный университет"
Елабужский институт (филиал)

**Фонд оценочных средств по дисциплине (модулю)
Б1.О.08.05 Программирование и робототехника**

Направление подготовки: 44.03.05 - Педагогическое образование (с двумя профилями подготовки)
Профиль подготовки: Математика и информатика
Квалификация выпускника: бакалавр
Форма обучения: очное
Язык обучения: русский
Год начала обучения по образовательной программе: 2023

СОДЕРЖАНИЕ

- [1. Соответствие компетенций планируемым результатам обучения по дисциплине \(модулю\)](#)
- [2. Критерии оценивания сформированности компетенций](#)
- [3. Распределение оценок за формы текущего контроля и промежуточную аттестацию](#)
- [4. Оценочные средства, порядок их применения и критерии оценивания](#)
 - [4.1. Оценочные средства текущего контроля](#)
 - [4.1.1. Лабораторные работы](#)
 - [4.1.1.1. Порядок проведения.](#)
 - [4.1.1.2 Критерии оценивания](#)
 - [4.1.1.3. Содержание оценочного средства](#)
 - [4.1.2. Контрольная работа](#)
 - [4.1.2.1. Порядок проведения.](#)
 - [4.1.2.2 Критерии оценивания](#)
 - [4.1.2.3. Содержание оценочного средства](#)
 - [4.2. Оценочные средства промежуточной аттестации \(экзамен\)](#)
 - [4.2.1. Устный или письменный ответ на вопрос](#)
 - [4.2.1.1. Порядок проведения.](#)
 - [4.2.1.2. Критерии оценивания.](#)
 - [4.2.1.3. Оценочные средства.](#)
 - [4.2.2. Практическое задание](#)
 - [4.2.2.1. Порядок проведения.](#)
 - [4.2.2.2. Критерии оценивания.](#)
 - [4.2.2.3. Оценочные средства.](#)

1. Соответствие компетенций планируемым результатам обучения по дисциплине (модулю)

Код и наименование компетенции	Индикаторы достижения компетенций для данной дисциплины	Оценочные средства текущего контроля и промежуточной аттестации
<p>УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач</p>	<p>Знать эффективные технологии организации контроля и оценки формирования результатов образования обучающихся, выявления и коррекции трудностей в обучении программированию и робототехнике</p> <p>Уметь выбирать и применять эффективные технологии организации контроля и оценки формирования результатов образования обучающихся, выявления и коррекции трудностей в обучении программированию и робототехнике</p> <p>Владеть эффективными технологиями осуществления контроля и оценки формирования результатов образования обучающихся, выявления и коррекции трудностей в обучении, возникающих при решении задач программирования</p>	<p>Текущий контроль: Лабораторные работы по темам Тема 1. Программирование Arduino. Введение. Структура программы, константы. Цифровой ввод/вывод. Аналоговый ввод/вывод. Тема 2. Дополнительные функции ввода/вывода. Работа со временем. Математические функции. Псевдослучайные числа Тема 3. Последовательная передача данных. Прерывания. EEPROM. Blink без delay. Тема 4. Создание библиотеки. Ethernet, Servo, Firmata library. Работа с библиотеками. Тема 5. Цифровой ввод - кнопка. Аналоговый вывод - Fading.</p> <p>Контрольная работа по темам Тема 6. Аналоговый ввод - потенциометр и осциллограф. Тема 7. Генерация звука - пьезоизлучатель. Фоторезистор. Сенсор на светодиоде. Общение с Arduino - программирование работы с COM-портом. Подключаем к Arduino мышку PS/2</p> <p>Промежуточная аттестация: <i>Зачет, Экзамен</i></p>
<p>УК-2. Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений</p>	<p>Знать требования к определению круга задач в рамках поставленной цели; способы эффективного решения задач программированию и робототехнике с учетом действующих правовых норм, имеющихся ресурсов и ограничений</p> <p>Уметь определять круг задач программированию и робототехнике задач в рамках поставленной цели, выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений</p> <p>Владеть навыками определения круга задач по программированию и робототехнике в рамках поставленной цели, выбора оптимальных способов их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений</p>	<p>Текущий контроль: Лабораторные работы по темам Тема 1. Программирование Arduino. Введение. Структура программы, константы. Цифровой ввод/вывод. Аналоговый ввод/вывод. Тема 2. Дополнительные функции ввода/вывода. Работа со временем. Математические функции. Псевдослучайные числа Тема 3. Последовательная передача данных. Прерывания. EEPROM. Blink без delay. Тема 4. Создание библиотеки. Ethernet, Servo, Firmata library. Работа с библиотеками. Тема 5. Цифровой ввод - кнопка. Аналоговый вывод - Fading.</p> <p>Контрольная работа по темам Тема 1. Программирование Arduino. Введение. Структура программы, константы. Цифровой ввод/вывод. Аналоговый ввод/вывод. Тема 2. Дополнительные функции ввода/вывода. Работа со временем. Математические функции. Псевдослучайные числа Тема 3. Последовательная передача данных.</p>

		<p>Прерывания. EEPROM. Blink без delay. Тема 4. Создание библиотеки. Ethernet, Servo, Firmata library. Работа с библиотеками. Тема 5. Цифровой ввод - кнопка. Аналоговый вывод - Fading.</p> <p>Промежуточная аттестация: <i>Зачет, Экзамен</i></p>
<p>ОПК-2 Способен участвовать в разработке основных и дополнительных образовательных программ, разрабатывать отдельные их компоненты (в том числе с использованием информационно-коммуникационных технологий)</p>	<p>Владеть навыками разработки в составе команды основных и дополнительных образовательных программ (в том числе с использованием программирования и робототехники) с учетом особенностей конкретной образовательной среды и индивидуальных потребностей обучающихся</p>	<p>Текущий контроль: Лабораторные работы по темам Тема 8. Аналоговый датчик температуры - LM335. Протокол 1-Wire и iButton. Arduino и эмулятор iButton. Arduino и температурный 1-Wire датчик DS18S20. Тема 9. Arduino и драйвер двигателей L293D (Простой мотор-шилд). Arduino и сервомашинка. LCD-дисплей на базе HD44780 и Arduino</p> <p>Контрольная работа по темам Тема 8. Аналоговый датчик температуры - LM335. Протокол 1-Wire и iButton. Arduino и эмулятор iButton. Arduino и температурный 1-Wire датчик DS18S20. Тема 9. Arduino и драйвер двигателей L293D (Простой мотор-шилд). Arduino и сервомашинка. LCD-дисплей на базе HD44780 и Arduino</p> <p>Промежуточная аттестация: <i>Зачет, Экзамен</i></p>

2. Критерии оценивания сформированности компетенций

Компетенция	Зачтено			Не зачтено
	Высокий уровень (отлично) (86-100 баллов)	Средний уровень (хорошо) (71-85 баллов)	Низкий уровень (удовлетворительно) (56-70 баллов)	Ниже порогового уровня (неудовлетворительно) (0-55 баллов)
УК-1	Знает эффективные технологии организации контроля и оценки формирования результатов образования обучающихся, выявления и коррекции трудностей в обучении программированию и робототехнике	Знает эффективные технологии организации контроля и оценки формирования результатов образования обучающихся, допускает незначительные ошибки в выявлении и коррекции трудностей в обучении программированию и робототехнике	Знает эффективные технологии организации контроля и оценки формирования результатов образования обучающихся, допускает типичные ошибки в выявлении и коррекции трудностей в обучении программированию и робототехнике	Не знает эффективные технологии организации контроля и оценки формирования результатов образования обучающихся, выявления и коррекции трудностей в обучении программированию и робототехнике
	Умеет выбирать и применять эффективные технологии организации контроля и оценки формирования результатов образования	Умеет выбирать и применять эффективные технологии организации контроля и оценки формирования результатов образования обучающихся, выявления и коррекции трудностей в обучении	Умеет выбирать и применять эффективные технологии организации контроля и оценки формирования результатов образования	Не умеет выбирать и применять эффективные технологии организации контроля и оценки формирования результатов образования обучающихся, выявления и коррекции трудностей в обучении

	ресурсов и ограничений	незначительные ошибки при решении поставленных задач	ресурсов и ограничений, допуская типичные ошибки при решении поставленных задач	
<i>ОПК-2</i>	Владеет навыками разработки в составе команды основных и дополнительных образовательных программ (в том числе с использованием программирования и робототехники) с учетом особенностей конкретной образовательной среды и индивидуальных потребностей обучающихся	Владеет навыками разработки в составе команды основных и дополнительных образовательных программ с использованием информационно-коммуникационных технологий с учетом особенностей применения аппаратных средств компьютера в конкретной образовательной среде, допуская несущественные затруднения	Владеет навыками разработки в составе команды основных и дополнительных образовательных программ с использованием основных информационно-коммуникационных технологий с учетом особенностей применения аппаратных средств, допуская типичные ошибки	Не владеет навыками разработки в составе команды основных и дополнительных образовательных программ (в том числе с использованием программирования и робототехники) с учетом особенностей конкретной образовательной среды и индивидуальных потребностей обучающихся

3. Распределение оценок за формы текущего контроля и промежуточную аттестацию

5 семестр:

Текущий контроль:

Лабораторные работы по темам

Тема 1. Программирование Arduino. Введение. Структура программы, константы. Цифровой ввод/вывод.

Аналоговый ввод/вывод.

Тема 2. Дополнительные функции ввода/вывода. Работа со временем. Математические функции. Псевдослучайные числа

Тема 3. Последовательная передача данных. Прерывания. EEPROM. Blink без delay.

Тема 4. Создание библиотеки. Ethernet, Servo, Firmata library. Работа с библиотеками.

Максимальное количество баллов по БРС - 30.

Контрольная работа Тема 1. Программирование Arduino. Введение. Структура программы, константы. Цифровой ввод/вывод. Аналоговый ввод/вывод.

Тема 2. Дополнительные функции ввода/вывода. Работа со временем. Математические функции. Псевдослучайные числа

Тема 3. Последовательная передача данных. Прерывания. EEPROM. Blink без delay.

Тема 4. Создание библиотеки. Ethernet, Servo, Firmata library. Работа с библиотеками.

Максимальное количество баллов по БРС - 20.

Итого 30+20=50 баллов

Промежуточная аттестация – зачет – 50 баллов.

8 семестр:

Текущий контроль:

Лабораторные работы по темам

Тема 5. Цифровой ввод - кнопка. Аналоговый вывод - Fading.

Тема 6. Аналоговый ввод - потенциометр и осциллограф.

Тема 7. Генерация звука - пьезоизлучатель. Фоторезистор. Сенсор на светодиоде. Общение с Arduino - программирование работы с COM-портом. Подключаем к Arduino мышку PS/2

Тема 8. Аналоговый датчик температуры - LM335. Протокол 1-Wire и iButton. Arduino и эмулятор iButton. Arduino и температурный 1-Wire датчик DS18S20.

Тема 9. Arduino и драйвер двигателей L293D (Простой мотор-шилд). Arduino и сервомашинка. LCD-дисплей на базе HD44780 и Arduino

Максимальное количество баллов по БРС - 30.

Контрольная работа Тема 5. Цифровой ввод - кнопка. Аналоговый вывод - Fading.

Тема 6. Аналоговый ввод - потенциометр и осциллограф.

Тема 7. Генерация звука - пьезоизлучатель. Фоторезистор. Сенсор на светодиоде. Общение с Arduino - программирование работы с COM-портом. Подключаем к Arduino мышку PS/2

Тема 8. Аналоговый датчик температуры - LM335. Протокол 1-Wire и iButton. Arduino и эмулятор iButton. Arduino и температурный 1-Wire датчик DS18S20.

Тема 9. Arduino и драйвер двигателей L293D (Простой мотор-шилд). Arduino и сервомашинка. LCD-дисплей на базе HD44780 и Arduino

Максимальное количество баллов по БРС - 20.

Итого 30+20=50 баллов

Промежуточная аттестация – экзамен – 50 баллов.

Промежуточная аттестация проводится после завершения изучения дисциплины или ее части в форме, определяемой учебным планом образовательной программы с целью оценить работу обучающегося, степень усвоения теоретических знаний, уровень сформированности компетенций.

Преподаватель, принимающий зачет/экзамен обеспечивает случайное распределение вариантов зачетных заданий между обучающимися с помощью билетов и/или с применением компьютерных технологий; вправе задавать обучающемуся дополнительные вопросы и давать дополнительные задания помимо тех, которые указаны в билете. Зачет/экзамен проводится по билетам. В каждом билете два оценочных средства: устный или письменный ответ на вопрос и практическое задание.

Устный или письменный ответ – 20 баллов.

Практическое задание – 30 баллов.

Итого 20+30=50 баллов.

Общее количество баллов по дисциплине за текущий контроль и промежуточную аттестацию: 50+50=100 баллов.

Соответствие баллов и оценок:

Для зачета

56-100 – зачтено.

0-55 – не зачтено.

Для экзамена

86-100 – отлично.

71-85 – хорошо.

56-70 – удовлетворительно.

0-55 – неудовлетворительно.

4. Оценочные средства, порядок их применения и критерии оценивания

4.1. Оценочные средства текущего контроля

4.1.1. Лабораторные работы

Тема 1. Программирование Arduino. Введение. Структура программы, константы. Цифровой ввод/вывод.

Аналоговый ввод/вывод.

Тема 2. Дополнительные функции ввода/вывода. Работа со временем. Математические функции. Псевдослучайные числа

Тема 3. Последовательная передача данных. Прерывания. EEPROM. Blink без delay.

Тема 4. Создание библиотеки. Ethernet, Servo, Firmata library. Работа с библиотеками.

Тема 5. Цифровой ввод - кнопка. Аналоговый вывод - Fading.

Тема 6. Аналоговый ввод - потенциометр и осциллограф.

Тема 7. Генерация звука - пьезоизлучатель. Фоторезистор. Сенсор на светодиоде. Общение с Arduino - программирование работы с COM-портом. Подключаем к Arduino мышку PS/2

Тема 8. Аналоговый датчик температуры - LM335. Протокол 1-Wire и iButton. Arduino и эмулятор iButton. Arduino и температурный 1-Wire датчик DS18S20.

Тема 9. Arduino и драйвер двигателей L293D (Простой мотор-шилд). Arduino и сервомашинка. LCD-дисплей на базе HD44780 и Arduino

Лабораторные работы проводятся в часы аудиторной работы.

Перед выполнением каждой работы студенты-бакалавры должны проработать соответствующий материал, используя конспекты теоретических занятий, периодические издания, учебно-методические пособия и учебники.

По окончании занятий студенты оформляют отчет по каждой работе, соблюдая следующую форму:

- Наименование темы;
- Цель работы;
- Задание и содержание выполненной работы,
- Письменные ответы на контрольные вопросы.
- Выводы по проделанной работе.
- Список использованных источников.

4.1.1.2 Критерии оценивания

26-30 баллов ставится, если обучающийся:

В ответе качественно раскрыл содержание темы. Ответ хорошо структурирован. Прекрасно освоен понятийный аппарат. Продемонстрирован высокий уровень понимания материала. Превосходное умение формулировать свои мысли, обсуждать дискуссионные положения.

21-25 баллов ставится, если обучающийся:

Основные вопросы темы раскрыл. Структура ответа в целом адекватна теме. Хорошо освоен понятийный аппарат. Продемонстрирован хороший уровень понимания материала. Хорошее умение формулировать свои мысли, обсуждать дискуссионные положения.

17-20 баллов ставится, если обучающийся:

Тему частично раскрыл. Ответ слабо структурирован. Понятийный аппарат освоен частично. Понимание отдельных положений из материала по теме. Удовлетворительное умение формулировать свои мысли, обсуждать дискуссионные положения.

0--16 баллов ставится, если обучающийся:

Тему не раскрыл. Понятийный аппарат освоен неудовлетворительно. Понимание материала фрагментарное или отсутствует. Неумение формулировать свои мысли, обсуждать дискуссионные положения.

4.1.1.3. Содержание оценочного средства

Лабораторная работа 1

Arduino программируется на специальном языке программирования – он основан на C/C ++, и позволяет использовать любые его функции. Строго говоря, отдельного языка Arduino не существует, как и не существует компилятора Arduino – написанные программы преобразуются (с минимальными изменениям) в программу на языке C/C++, и затем компилируются компилятором AVR-GCC. Так что фактически, используется специализированный для микроконтроллеров AVR вариант C/C++.

Разница заключается в том, что Вы получаете простую среду разработки, и набор базовых библиотек, упрощающих доступ к находящейся «на борту» микроконтроллера периферии.

Согласитесь, очень удобно начать работу с последовательным портом на скорости 9600 бит в секунду, сделав вызов одной строчкой:

```
Serial.begin(9600);
```

А при использовании «голового» C/C++ Вам бы пришлось разбираться с документацией на микроконтроллер, и вызывать нечто подобное:

```
UBRR0H = ((F_CPU / 16 + 9600 / 2) / 9600 - 1) >> 8;
```

```
UBRR0L = ((F_CPU / 16 + 9600 / 2) / 9600 - 1);
```

```
sbi(UCSR0B, RXEN0);
```

```
sbi(UCSR0B, TXEN0);
```

```
sbi(UCSR0B, RXCIE0);
```

Здесь кратко рассмотрены основные функции и особенности программирования Arduino. Если Вы не знакомы с синтаксисом языков C/C++, советуем обратиться к любой литературе по данному вопросу, либо Internet-источникам.

С другой стороны, все представленные примеры очень просты, и скорее всего у Вас не возникнет трудностей с пониманием исходных текстов и написанием собственных программ даже без чтения дополнительной литературы.

Более полная документация (на английском языке) представлена на официальном сайте проекта – <http://www.arduino.cc> . Там же есть форум, ссылки на дополнительные библиотеки и их описание.

По аналогии с описанием на официальном сайте проекта Arduino, под «портом» понимается контакт микроконтроллера, выведенный на разъем под соответствующим номером. Кроме того, существует порт последовательной передачи данных (COM-порт).

Структура программы

В своей программе Вы должны объявить две основных функции: setup() и loop().

Функция setup() вызывается один раз, после каждого включения питания или сброса платы Arduino. Используйте её, чтобы инициализировать переменные, установить режимы работы цифровых портов и т.д.

Функция loop() последовательно раз за разом исполняет команды, которые описаны в ее теле. Т.е. после завершения функции снова произойдет ее вызов.

Разберем простой пример:

```
void setup()      // начальные установки
```

```
{
```

```
  beginSerial(9600); // установка скорости работы серийного порта на 9600 бит/сек
```

```
  pinMode(3, INPUT); // установка 3-его порта на ввод данных
```

```
}
```

```
// Программа проверяет 3-ий порт на наличие на нём сигнала и посылает ответ в
```

```
// виде текстового сообщения на последовательный порт компьютера
```

```
void loop() // тело программы
```

```
{
```

```
  if (digitalRead(3) == HIGH) // условие на опрос 3го порта
```

```
    serialWrite('H');      // отправка сообщения в виде буквы «H» на COM-порт
```

```
else
serialWrite('L'); // отправка сообщения в виде буквы «L» на COM-порт
  delay(1000); // задержка 1 сек.
}
```

Константы

Константы – предопределенные значения. Они используются, чтобы делать программы более легкими для чтения. Мы классифицируем константы в группах.

Уровни сигналов порта HIGH и LOW

При чтении или записи к цифровому порту применимо только два возможных значения – порт может быть установлен как HIGH (высокий уровень) или LOW (низкий уровень).

Уровень HIGH соответствует 5 вольтам на выходе. При чтении значения на цифровом порте, начиная с 3 вольт и выше, микропроцессор воспримет это напряжение как HIGH. Эта константа представлена целым числом 1.

Уровень LOW соответствует 0 вольтам на выходе порта. При чтении значения на цифровом порте, начиная с 2 вольт и меньше, микропроцессор воспримет это напряжение как LOW. Эта константа представлена целым числом 0.

Таким образом, оба следующих вызова будут эквивалентны:

```
digitalWrite(13, HIGH); // можно так,
digitalWrite(13, 1); // а можно и так
```

Считается, что первый вариант более нагляден.

Лабораторная работа 2

Настройка цифровых портов на ввод (INPUT) и вывод (OUTPUT) сигналов

Цифровые порты могут использоваться на ввод или вывод сигналов.

Изменение порта с ввода на вывод производится при помощи функции pinMode().

Порты, сконфигурированные на ввод сигналов, имеют большое входное сопротивление, что позволяет подключать к ним источник сигнала, и порт не будет потреблять большой ток.

Порты, сконфигурированные на вывод сигналов, имеют малое выходное сопротивление. Это означает, что такие порты могут обеспечивать подключенные к ним элементы электроэнергией. В этом состоянии порты поддерживают положительное или отрицательное направление тока до 40 мА (миллиампер) на другие устройства или схемы. Это позволяет подключить к ним какую-либо нагрузку, например светодиод (через резистор, ограничивающий ток). Порты, сконфигурированные как выводы, могут быть повреждены, если их замкнуть накоротко на «землю» (общая шина питания), на источник питания +5 В, или подсоединить к мощной нагрузке с малым сопротивлением.

Пример:

```
pinMode(13, OUTPUT); //13й вывод будет выходом
pinMode(12, INPUT); //а 12й – входом
```

Специфичные для Arduino функции и объекты

Цифровой ввод/вывод

pinMode

Вызов:

```
pinMode (порт, режим);
```

Описание:

Конфигурирует указанный порт на ввод или вывод сигнала.

Параметры:

порт – номер порта, режим которого Вы желает установить (значение целого типа от 0 до 13).

режим – либо INPUT (ввод) либо OUTPUT (вывод).

Пример:

```
pinMode(13, OUTPUT); //13й вывод будет выходом
pinMode(12, INPUT); //а 12й – входом
```

Примечание:

Аналоговые входы могут использоваться как цифровые входы/выходы, при обращении к ним по номерам с 14 (аналоговый вход 0) по 19 (аналоговый вход 5)

digitalWrite

Вызов:

```
digitalWrite(порт, значение);
```

Описание:

Устанавливает высокий (HIGH) или низкий (LOW) уровень напряжения на указанном порту.

Параметры:

порт: номер порта

значение: HIGH или LOW

Пример:

```
digitalWrite(13, HIGH); // выставляем 13й вывод в «высокое» состояние
```

digitalRead

Вызов:

```
value = digitalRead (порт);
```

Описание:

Считывает значение на указанном порту

Параметры:

порт: номер опрашиваемого порта

Возвращаемое значение: возвращает текущее значение на порту (HIGH или LOW) типа int

Пример:

```
int val;
```

```
val = digitalRead(12); // опрашиваем 12й вывод
```

Примечание:

Если к считываемому порту ничего не подключено, то функция digitalRead () может беспорядочно возвращать значения HIGH или LOW.

Лабораторная работа 3

Аналоговый ввод/вывод сигнала

analogRead

Вызов:

```
value = analogRead(порт);
```

Описание:

Считывает значение с указанного аналогового порта. Arduino содержит 6 каналов, аналого-цифрового преобразователя на 10 битов каждый. Это означает, что входное напряжения от 0 до 5В преобразовывается в целочисленное значение от 0 до 1023. Разрешающая способность считывания составляет: 5 В/1024 значений = 0,004883 В/значение (4,883 мВ). Требуется приблизительно 100 нС (0.0001 С), чтобы считать значение аналогового ввода, так что максимальная скорость считывания - приблизительно 10000 раз в секунду.

Параметры:

порт: номер опрашиваемого аналогового входа

Возвращаемое значение: возвращает число типа int в диапазоне от 0 до 1023, считанное с указанного порта.

Пример:

```
int val;
```

```
val = analogRead(0); // считываем значение на 0м аналоговом входе
```

Примечание:

Аналоговые порты по умолчанию определены на ввод сигнала и в отличие от цифровых портов их не требуется конфигурировать с помощью вызова функции pinMode.

analogWrite

Вызов:

```
analogWrite(порт, значение);
```

Описание:

Выводит на порт аналоговое значение. Эта функция работает на: 3, 5, 6, 9, 10, и 11 цифровых портах Arduino.

Может применяться для изменения яркости светодиода, для управления двигателем и т.д. После вызова функции analogWrite, соответствующий порт начинает работать в режиме широтно-импульсного модулирования напряжения до тех пор, пока не будет следующего вызова функции analogWrite (или функций digitalWrite / digitalWrite на том же самом порте).

Параметры:

порт: номер опрашиваемого аналогового входа

значение: целочисленное между 0 и 255. Значение 0 генерирует 0 В на указанном порте; значение 255 генерирует +5 В на указанном порте. Для значений между 0 и 255, порт начинает быстро чередовать уровень напряжения 0 и +5 В - чем выше значение, тем, более часто порт генерирует уровень HIGH (5 В).

Пример:

```
analogWrite(9, 128); // устанавливаем на 9 контакте значение эквивалентное 2,5В
```

Примечание:

Нет необходимости вызвать функцию pinMode, чтобы установить порт на вывод сигналов перед вызовом функции analogWrite.

Частота генерирования сигнала – приблизительно 490 Гц.

Лабораторная работа 4

Работа со временем

millis

Вызов:

```
time = millis();
```

Описание:

Возвращает число миллисекунд, с момента исполнения Arduino текущей программы. Счетчик переполнится и обнулится приблизительно через 9 часов.

Возвращаемое значение: возвращает значение типа unsigned long

Пример:

```
unsigned long time; // объявление переменной time типа unsigned long  
time = millis(); // передача количества миллисекунд
```

delay

Вызов:

```
delay(время_мс);
```

Описание:

Приостанавливает программу на заданное число миллисекунд.

Параметры:

время_мс – время задержки программы в миллисекундах

Пример:

```
delay(1000); //пауза 1 секунда
```

delayMicroseconds

Вызов:

```
delayMicroseconds(время_мкс);
```

Описание:

Приостанавливает программу на заданное число микросекунд.

Параметры:

время_мкс – время задержки программы в микросекундах

Пример:

```
delayMicroseconds(500); //пауза 500 микросекунд
```

pulseIn

Вызов:

```
pulseIn(порт, значение);
```

Описание:

Считывает импульс (высокий или низкий) с цифрового порта и возвращает продолжительность импульса в микросекундах.

Например, если параметр «значение» при вызове функции установлен в HIGH, то pulseIn() ожидает, когда на порт поступит высокий уровень сигнала. С момента его поступления начинается отсчет времени до тех пор, пока на порт не поступит низкий уровень сигнала. Функция возвращает длину импульса (высокого уровня) в микросекундах. Работает с импульсами от 10 микросекунд до 3 минут. Обратите внимание, что эта функция не будет возвращать результат, пока импульс не будет обнаружен.

Параметры:

порт: номер порта, с которого считываем импульс

значение: тип импульса HIGH или LOW

Возвращаемое значение: возвращает длительность импульса в микросекундах (тип int)

Пример:

```
int duration; // объявление переменной duration типа int  
duration = pulseIn(pin, HIGH); // измеряем длительность импульса
```

Лабораторная работа 5

Последовательная передача данных

Arduino имеет встроенный контроллер для последовательной передачи данных, который может использоваться как для связи между Arduino/Arduino устройствами, так и для связи с компьютером. На компьютере соответствующее соединение представлено USB COM-портом.

Связь происходит по цифровым портам 0 и 1, и поэтому Вы не сможете использовать их для цифрового ввода/вывода если используете функции последовательной передачи данных.

Serial.begin

Вызов:

```
Serial.begin(скорость_передачи);
```

Описание:

Устанавливает скорость передачи информации COM порта битах в секунду для последовательной передачи данных. Для того чтобы поддерживать связь с компьютером, используйте одну из этих нормированных скоростей: 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, или 115200. Также Вы можете определить другие скорости при связи с другим микроконтроллером по портам 0 и 1.

Параметры:

скорость_передачи: скорость потока данных в битах в секунду.

Пример:

```
Serial.begin(9600); //устанавливаем скорость 9600 бит/сек
```

Serial.available

Вызов:

```
count = Serial.available();
```

Описание:

Принимаемые по последовательному порту байты попадают в буфер микроконтроллера, откуда Ваша программа может их считать. Функция возвращает количество накопленных в буфере байт. Последовательный буфер может хранить до 128 байт.

Возвращаемое значение:

Возвращает значение типа `int` – количество байт, доступных для чтения, в последовательном буфере, или 0, если ничего не доступно.

Пример:

```
if (Serial.available() > 0) { // Если в буфере есть данные
// здесь должен быть прием и обработка данных
}
```

Serial.read

Вызов:

```
char = Serial.read();
```

Описание:

Считывает следующий байт из буфера последовательного порта.

Возвращаемое значение:

Первый доступный байт входящих данных с последовательного порта, или -1 если нет входящих данных.

Пример:

```
incomingByte = Serial.read(); // читаем байт
```

Serial.flush

Вызов:

```
Serial.flush();
```

Описание:

Очищает входной буфер последовательного порта. Находящиеся в буфере данные теряются, и дальнейшие вызовы `Serial.read()` или `Serial.available()` будут иметь смысл для данных, полученных после вызова `Serial.flush()`.

Пример:

```
Serial.flush(); // Очищаем буфер – начинаем прием данных «с чистого листа»
```

Serial.print

Описание:

Вывод данных на последовательный порт.

Параметры:

Функция имеет несколько форм вызова в зависимости от типа и формата выводимых данных.

`Serial.print(b, DEC)` выводит ASCII-строку - десятичное представление числа `b`.

```
int b = 79;
```

```
Serial.print(b, DEC); //выдаст в порт строку «79»
```

`Serial.print(b, HEX)` выводит ASCII-строку - шестнадцатеричное представление числа `b`.

```
int b = 79;
```

```
Serial.print(b, HEX); //выдаст в порт строку «4F»
```

`Serial.print(b, OCT)` выводит ASCII-строку - восьмеричное представление числа `b`.

```
int b = 79;
```

```
Serial.print(b, OCT); //выдаст в порт строку «117»
```

`Serial.print(b, BIN)` выводит ASCII-строку - двоичное представление числа `b`.

```
int b = 79;
```

```
Serial.print(b, BIN); //выдаст в порт строку «1001111»
```

`Serial.print(b, BYTE)` выводит младший байт числа `b`.

```
int b = 79;
```

```
Serial.print(b, BYTE); //выведет число 79 (один байт). В мониторе
```

```
//последовательного порта получим символ «O» - его
```

```
//код равен 79
```

`Serial.print(str)` если `str` – строка или массив символов, побайтно передает `str` на COM-порт.

```
char bytes[3] = {79, 80, 81}; //массив из 3 байт со значениями 79,80,81
```

```
Serial.print("Here our bytes:"); //выводит строку «Here our bytes:»
```

```
Serial.print(bytes); //выводит 3 символа с кодами 79,80,81 –
```

```
//это символы «OPQ»
```

`Serial.print(b)` если `b` имеет тип `byte` или `char`, выводит в порт само число `b`.

```
char b = 79;
```

```
Serial.print(b); //выдаст в порт символ «O»
```

`Serial.print(b)` если `b` имеет целый тип, выводит в порт десятичное представление числа `b`.

```
int b = 79;
```

```
Serial.print(b); //выдаст в порт строку «79»
```

Serial.println

Описание:

Функция Serial.println аналогична функции Serial.print, и имеет такие же варианты вызова. Единственное отличие заключается в том, что после данных дополнительно выводятся два символа – символ возврата каретки (ASCII 13, или '\r') и символ новой линии (ASCII 10, или '\n').

Пример 1 и пример 2 выведут в порт одно и то же:

Пример 1:

```
int b = 79;
```

```
Serial.print(b, DEC); //выдаст в порт строку «79»
```

```
Serial.print("\r\n"); //выведет символы "\r\n" – перевод строки
```

```
Serial.print(b, HEX); //выдаст в порт строку «4F»
```

```
Serial.print("\r\n");//выведет символы "\r\n" – перевод строки
```

Пример 2:

```
int b = 79;
```

```
Serial.println(b, DEC); //выдаст в порт строку «79\r\n»
```

```
Serial.println(b, HEX); //выдаст в порт строку «4F\r\n»
```

В мониторе последовательного порта получим:

```
79
```

```
4F
```

4.1.2. Контрольная работа

Тема 1. Программирование Arduino. Введение. Структура программы, константы. Цифровой ввод/вывод.

Аналоговый ввод/вывод.

Тема 2. Дополнительные функции ввода/вывода. Работа со временем. Математические функции. Псевдослучайные числа

Тема 3. Последовательная передача данных. Прерывания. EEPROM. Blink без delay.

Тема 4. Создание библиотеки. Ethernet, Servo, Firmata library. Работа с библиотеками.

Тема 5. Цифровой ввод - кнопка. Аналоговый вывод - Fading.

Тема 6. Аналоговый ввод - потенциометр и осциллограф.

Тема 7. Генерация звука - пьезоизлучатель. Фоторезистор. Сенсор на светодиоде. Общение с Arduino - программирование работы с COM-портом. Подключаем к Arduino мышку PS/2

Тема 8. Аналоговый датчик температуры - LM335. Протокол 1-Wire и iButton. Arduino и эмулятор iButton. Arduino и температурный 1-Wire датчик DS18S20.

Тема 9. Arduino и драйвер двигателей L293D (Простой мотор-шилд). Arduino и сервомашинка. LCD-дисплей на базе HD44780 и Arduino

4.1.2.1. Порядок проведения.

Контрольная работа проходит в форме тестирования. Тестирование проходит в письменной форме или с использованием компьютерных средств. Обучающийся получает определённое количество тестовых заданий. На выполнение выделяется фиксированное время в зависимости от количества заданий. Оценка выставляется в зависимости от процента правильно выполненных заданий. Тестирование проводится по вариантам. В каждом варианте – 10 тестовых заданий. За каждый правильный ответ начисляется 1 балл. Итого за тестирование студент может заработать до 10 баллов.

Ниже приведены примерные задания. Полный банк тестовых заданий хранится на кафедре.

4.1.2.2 Критерии оценивания

16-20 баллов ставится, если обучающийся:

86% правильных ответов и более.

11-15 баллов ставится, если обучающийся:

От 71% до 85 % правильных ответов.

6-10 баллов ставится, если обучающийся:

От 56% до 70% правильных ответов.

0--5 балла ставится, если обучающийся:

55% правильных ответов и менее.

4.1.2.3. Содержание оценочного средства

1. Какое напряжение будет на датчике, если ТКУ = 10 мВ/К, а температура -23 °С

A. 2.5015V

B. 2.9615V

- C. 2.015V
 - D. 2.1V
2. Аналоговый датчик температуры LM335 может мерить температуру в диапазоне
 - A. от -40 °C до +100°C
 - B. от -50 °C до +200°C
 - C. от -60 °C до +1000°C
 - D. от -80 °C до +200°C
 3. Аналоговый датчик температуры LM335 может мерить температуру с точностью
 - A. До 1 градуса Цельсия
 - B. До 0,1 градуса Цельсия
 - C. До 0,5 градуса Цельсия
 - D. До 2 градусов Цельсия
 4. Что такое 1-Wire
 - A. это однопроводной интерфейс
 - B. это двухпроводной интерфейс
 - C. это датчик
 - D. это провод
 5. Что такое DS18S20
 - A. это высокоточный цифровой термометр с интерфейсом 1-Wire
 - B. это высокоточный цифровой барометр с интерфейсом 1-Wire
 - C. это высокоточный цифровой вольтметр с интерфейсом 1-Wire
 - D. это высокоточный цифровой амперметр с интерфейсом 1-Wire
 6. Датчик DS18S20 может мерить температуру в диапазоне
 - A. от -55 до +125 °C
 - B. от -65 до +135 °C
 - C. от -75 до +145 °C
 - D. от -85 до +155 °C
 7. Датчик DS18S20 может мерить температуру с точностью **0.5 °C** в диапазоне
 - A. от -10 до +85 °C
 - B. от -20 до +95 °C
 - C. от -30 до +105 °C
 - D. от -40 до +115 °C
 8. Какое максимальное время уходит на конвертацию температуры в DS18S20
 - A. 750 ms
 - B. 850 ms
 - C. 950 ms
 - D. 650 ms
 9. Сколько бит содержит ROM датчика DS18S20
 - A. 64
 - B. 16
 - C. 32
 - D. 56

Вопрос 10. Память датчика DS18S20 состоит из (выбрать из возможных вариантов)

- A. оперативной ROM
- B. энергонезависимой EEPROM
- C. флеш-памяти
- D. короткой памяти

Вопрос 11. Максимальная нагрузка на порт МК – не более...(40 мА)

Вопрос 12. Из кода представленного ниже скажите, к какому порту подключен аналоговый прибор

- ```
1. int photoPin = 1;
2. int ledPin = 13;
```

```
3. int val = 0;
4. void setup()
5. {
6. Serial.begin(9600);
7. }
8. void loop()
9. {
10. digitalWrite(ledPin, HIGH);
11. val = analogRead(photoPin);
12. Serial.println(val);
13. delay(200);
14. }
```

- A. К порту A1
- B. К порту A0
- C. К порту A2
- D. К порту A3

Вопрос 13. В какой строке кода считываются показания с аналогового прибора

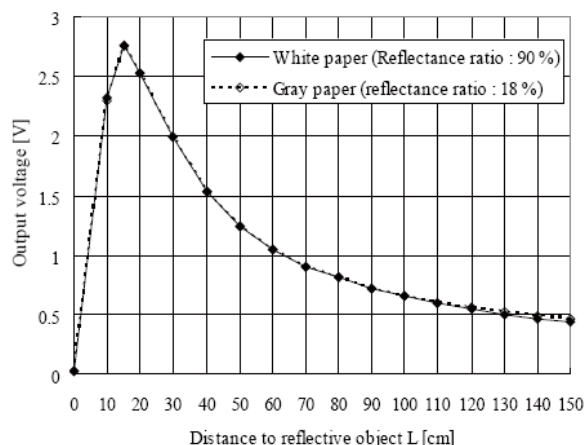
```
1. int photoPin = 1;
2. int ledPin = 13;
3. int val = 0;
4. void setup()
5. {
6. Serial.begin(9600);
7. }
8. void loop()
9. {
10. digitalWrite(ledPin, HIGH);
11. val = analogRead(photoPin);
12. Serial.println(val);
13. delay(200);
14. }
```

- A. В строке 11
- B. В строке 12
- C. В строке 10
- D. В строке 6

Вопрос 14. Сенсор SHARP-GP2Y0A02YK0F имеет диапазон измерения расстояния:

- A. от 20 до 150 см
- B. от 10 до 250 см
- C. от 20 до 250 см
- D. от 30 до 250 см

Вопрос 15. На графике представлена зависимость расстояния от напряжения сенсора SHARP-GP2Y0A02YK0F



Чем объяснить излом в начале графика?

- A. неспособностью дальномера обнаруживать объекты на близком расстоянии
- B. чувствительностью дальномера
- C. способностью дальномера обнаруживать объекты на близком расстоянии
- D. технической особенностью дальномера

Вопрос 16. В какой строке кода выводятся показания с аналогового прибора в монитор порта

```
1. int IRpin = 0;
2. void setup() {
3. Serial.begin(9600);
4. }
5. void loop() {
6. float volts = analogRead(IRpin)*0.0048828125;
7. float distance = 65*pow(volts, -1.10);
8. Serial.println(distance);
9. delay(100);
10. }
```

- A. В строке 8
- B. В строке 6
- C. В строке 7
- D. В строке 3

Вопрос 17. Ультразвуковой датчик измерения расстояния HC-SR04 (Ultrasonic ranging module HC-SR04) имеет эффективный угол:

- A. Менее 15°
- B. Более 15°
- C. Равный 15°
- D. нестабильный

Вопрос 18. Ультразвуковой датчик измерения расстояния **HC-SR04** (Ultrasonic ranging module HC-SR04) имеет диапазон измерения расстояния:

- A. 2–400 cm
- B. 1–400 cm
- C. 2–200 cm
- D. 2–100 cm

Вопрос 19. Ультразвуковой датчик измерения расстояния **HC-SR04** (Ultrasonic ranging module HC-SR04) имеет разрешение

- A. 0.3 cm
- B. 0.4 cm
- C. 0.5 cm
- D. 0.6 cm

Вопрос 20. В какой строке кода указано к каким портам подключен датчик Ultrasonic:

```
1. #include "Ultrasonic.h"
2. Ultrasonic ultrasonic(12, 13);
3. void setup()
4. {
5. Serial.begin(9600);
6. }
7. void loop()
8. {
9. float dist_cm = ultrasonic.Ranging(CM);
10. Serial.println(dist_cm);
11. delay(100);
12. }
```

- A. В строке 2
- B. В строке 5
- C. В строке 9
- D. В строке 10

Вопрос 21. Какой допустимый ток нагрузки у микросхемы L293D

- A. допустимый ток нагрузки 600мА (на каждый канал)
- B. допустимый ток нагрузки 800мА (на каждый канал)
- C. допустимый ток нагрузки 1000мА (на каждый канал)
- D. допустимый ток нагрузки 1200мА (на каждый канал)

Вопрос 22.

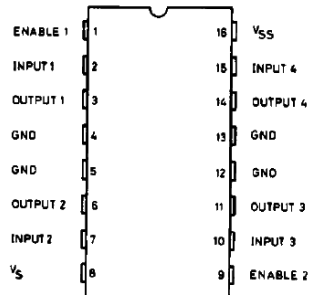
Какой допустимый ток нагрузки у микросхемы L293E

- A. допустимый ток нагрузки 1000мА (на каждый канал)
- B. допустимый ток нагрузки 800мА (на каждый канал)

- C. допустимый ток нагрузки 600мА (на каждый канал)
- D. допустимый ток нагрузки 1200мА (на каждый канал)

Вопрос 23.

Для управления скоростью вращения моторчиков используют ножки



- A. ENABLE1 и ENABLE2.
- B. INPUT1, INPUT2
- C. OUTPUT1, OUTPUT2
- D. INPUT1, INPUT2, INPUT3, INPUT4
- E.

Вопрос 24.

Для управления униполярным шаговым двигателем 28byj-48 можно использовать микросхему

- A. ULN2003
- B. L293D
- C. L293E
- D. L298

Вопрос 25.

Шаговые двигатели бывают

- A. униполярные
- B. биполярные
- C. полуполярные
- D. среднеполярные

Вопрос 26.

В какой строчке кода мы задаем скорость вращения шагового двигателя

1. `#include <Stepper.h>`
2. `const int IN1 = 2;`
3. `const int IN2 = 3;`
4. `const int IN3 = 4;`
5. `const int IN4 = 5;`
6. `const int stepsPerRevolution = 32;`
7. `Stepper myStepper(stepsPerRevolution, IN1, IN2, IN3, IN4);`
8. `void setup() {`

```
9. myStepper.setSpeed(5);
10. }
11. void loop() {
12. myStepper.step(stepsPerRevolution);
13. delay(500);
14. Serial.println("counterclockwise");
15. myStepper.step(-stepsPerRevolution);
16. delay(500);
17. }
```

- A. В строке 9
- B. В строке 6
- C. В строке 12
- D. В строке 15

Вопрос 27.

Этот дисплей, как и прочие на контроллере HD44780, поддерживает **...(вставить 2)** варианта параллельного интерфейса:

Вопрос 28. В какой строчке кода мы проводим инициализацию дисплея

```
1. #include <LiquidCrystal.h>
2. LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
3. void setup()
4. {
5. lcd.begin(16, 2);
6. lcd.print("hello, world!");
7. }
8. void loop()
9. {
10. lcd.setCursor(0, 1);
11. lcd.print(millis() / 1000);
12. }
```

- A. В строке 5
- B. В строке 6
- C. В строке 2

D. В строке 10

Ответы:

- 1) 1;
- 2) 4;
- 3) 4;
- 4) 4;
- 5) 1;
- 6) 2;
- 7) 3;
- 8) 6;
- 9) 4;
- 10) 5;
- 11) 4;
- 12) 3;
- 13) 2;
- 14) 2;
- 15) 3;
- 16) 3;
- 17) 3;
- 18) 5;
- 19) 3;
- 20) 1; 4;
- 21) 2;
- 22) 4;
- 23) 2;
- 24) 1;
- 25) 1;
- 26) 2;
- 27) 3;
- 28) 4.

## 4.2. Оценочные средства промежуточной аттестации

### 4.2.1. Устный ответ

#### 4.2.1.1. Порядок проведения.

Промежуточная аттестация нацелена на комплексную проверку освоения дисциплины. Обучающийся получает вопрос(ы)/задание(я) и время на подготовку. Промежуточная аттестация проводится в устной, письменной или компьютерной форме. Оценивается владение материалом, его системное освоение, способность применять нужные знания, навыки и умения при анализе проблемных ситуаций и решении практических заданий.

#### 4.2.1.2. Критерии оценивания.

Устный или письменный ответ на вопрос направлен на проверку знаний основных разделов по дисциплине.

##### 4.2.1.2. Критерии оценивания.

###### **18-20 баллов ставится, если обучающийся:**

В ответе качественно раскрыл содержание темы. Ответ хорошо структурирован. Прекрасно освоен понятийный аппарат. Продемонстрирован высокий уровень понимания материала. Превосходное умение формулировать свои мысли, обсуждать дискуссионные положения.

###### **14-17 баллов ставится, если обучающийся:**

Основные вопросы темы раскрыл. Структура ответа в целом адекватна теме. Хорошо освоен понятийный аппарат. Продемонстрирован хороший уровень понимания материала. Хорошее умение формулировать свои мысли, обсуждать дискуссионные положения.

###### **11-13 баллов ставится, если обучающийся:**

Тему частично раскрыл. Ответ слабо структурирован. Понятийный аппарат освоен частично. Понимание отдельных положений из материала по теме. Удовлетворительное умение формулировать свои мысли, обсуждать дискуссионные положения.

#### **0--10 баллов ставится, если обучающийся:**

Тему не раскрыл. Понятийный аппарат освоен неудовлетворительно. Понимание материала фрагментарное или отсутствует. Неумение формулировать свои мысли, обсуждать дискуссионные положения.

#### **4.2.1.3. Оценочные средства.**

##### **Семестр 5**

Вопросы к зачету:

1. Аналоговый ввод - потенциометр и осциллограф.
2. Генерация звука - пьезоизлучатель. Фоторезистор.
3. Сенсор на светодиоде. Программирование работы с COM-портом.
4. Аналоговый датчик температуры, LM335.
5. Протокол 1-Wire и iButton. Arduino и эмулятор iButton.
6. Arduino и температурный 1-Wire датчик DS18S20.
7. Arduino и драйвер двигателей L293D (Простой мотор-шилд).
8. Arduino и сервомашинка.
9. LCD-дисплей на базе HD44780 и Arduino.
10. ИК-датчик препятствий для Arduino на базе фототранзистора

##### **Семестр 8**

Вопросы к экзамену:

1. Структура программы, константы.
2. Цифровой ввод/вывод. Аналоговый ввод/вывод.
3. Дополнительные функции ввода/вывода. Работа со временем.
4. Математические функции. Псевдослучайные числа
5. Последовательная передача данных.
6. Прерывания. EEPROM.
7. Blink без delay.
8. Создание библиотеки.
9. Ethernet, Servo, Firmata library, работа с библиотеками.
10. Цифровой ввод, кнопка. Аналоговый вывод, Fading.

#### **4.2.2. Практическое задание**

##### **4.2.2.1. Порядок проведения.**

Практические навыки проверяются путём выполнения обучающимися практических заданий в условиях, полностью или частично приближенных к условиям профессиональной деятельности. Проверяется знание теоретического материала, необходимое для правильного совершения необходимых действий, умение выстроить последовательность действий, практическое владение приёмами и методами решения профессиональных задач.

##### **4.2.2.2. Критерии оценивания**

**27-30 баллов ставится, если обучающимся:**

Задание выполнено полностью и правильно.

**22-26 баллов ставится, если обучающимся:**

Задание выполнено полностью, но нет достаточного обоснования. Или при верном решении допущена ошибка или недочет, не влияющий на правильную последовательность рассуждений.

**18-21 баллов ставится, если обучающимся:**

Задание выполнено частично или с фактическими ошибками.

**0-17 баллов ставится, если обучающимся:**

Задание не выполнено или выполнено с большим количеством фактических ошибок.

##### **4.2.2.3. Оценочные средства.**

##### **Семестр 5**

1. Написать программу для работы с микросхемой 74HC595 (управление яркостью произвольного светодиода)
2. Собрать схему из 8 светодиодов и написать программу для их управления с помощью потенциометра
3. Написать программу для проигрывания мелодии в тональном режиме
4. Написать программу для зажигания (blink) светодиода, в зависимости от количества нажатий на кнопку
5. Написать программу для вывода 10 разрядных чисел (в случайном порядке) на 7-ми сегментный индикатор
6. Написать программу для вывода круговой «бегущей» свето-индикации на 7-ми сегментном индикаторе
7. Написать программу для считывания показаний с фоторезистора
8. Написать программу для управления двигателем DC (плавный пуск и остановка, регулирование

оборотов)

9. Написать программу для управления сервомашинкой (контроль угла поворота вала)
10. Написать программу для работы с микросхемой 74НС595 (каскадное подключение и зажигание 16 светодиодов в произвольном порядке)
11. Написать программу для работы с микросхемой 74НС595 (создать эффект «жалюзи»)

#### **Семестр 8**

1. Написать программу управления яркостью светодиода, используя последовательный интерфейс Ардуино
2. Написать программу прошивки «чистого» микроконтроллера AVR
3. Написать программу калькулятор с обработкой прерываний
4. Написать программу кодовый замок, используя функции attachInterrupt и DetachInterrupt
5. Написать программу массив данных, сохранив элементы в EEPROM Ардуино
6. Написать программу будильник, для микроконтроллера ATmega328
7. Написать программу светофор для пешехода
8. Написать программу «умный светильник»
9. Написать программу медиаплеер
10. Написать программу «записная книжка»

**Перечень литературы, необходимой для освоения дисциплины (модуля)**

Направление подготовки: 44.03.05 - Педагогическое образование (с двумя профилями подготовки)

Профиль подготовки: Математика и информатика

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2023

**Основная литература:**

1. Мамичев, Д. Программирование на Ардуино. От простого к сложному / Мамичев Д. - Москва : СОЛОН-ПРЕСС, 2018. - 244 с. - ISBN 978-5-91359-292-7. - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <https://www.studentlibrary.ru/book/ISBN9785913592927.html>
2. Иванов, А. А. Основы робототехники : учебное пособие / А.А. Иванов. — 2-е изд., испр. — Москва : ИНФРА-М, 2022. — 223 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-012765-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1842546>
3. Программирование. Сборник задач : учебное пособие для вузов / О. Г. Архипов, В. С. Батасова, П. В. Гречкина [и др.] ; Под редакцией М. М. Марана. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 140 с. — ISBN 978-5-507-44322-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/223418>
4. Arduino. . Полный учебный курс. От игры к инженерному проекту : руководство / А. А. Салахова, О. А. Феоктистова, Н. А. Александрова, М. В. Храмова. — Москва : Лаборатория знаний, 2020. — 178 с. — ISBN 978-5-00101-886-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/151485>
5. Петин, В. В. 77 проектов для Arduino : учебно-методическое пособие / В. В. Петин. — Москва : ДМК Пресс, 2020. — 356 с. — ISBN 978-5-97060-697-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131676>
6. Петин, В. В. Практическая энциклопедия Arduino : энциклопедия / В. В. Петин, А. А. Биняковский. — 2-ое изд., испр. и доп. — Москва : ДМК Пресс, 2020. — 166 с. — ISBN 978-5-97060-798-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131675>

**Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем**

Направление подготовки: 44.03.05 - Педагогическое образование (с двумя профилями подготовки)

Профиль подготовки: Математика и информатика

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2023

Освоение дисциплины (модуля) предполагает использование следующего программного обеспечения и информационно-справочных систем:

Office Professional Plus 2010

GIMP

Inkscape

Notepad ++

Python

Lazarus

Электронная библиотечная система «ZNANIUM.COM»

Электронная библиотечная система Издательства «Лань»

Электронная библиотечная система «Консультант студента»