Горчаков Л.В., Колесников Д.А., Печерицин А.А.

ИСПОЛЬЗОВАНИЕ ОБЛАЧНЫХ ТЕХНОЛОГИЙ В ДИСТАНЦИОННОМ ЛАБОРАТОРНОМ ЭКСПЕРИМЕНТЕ

Создание удаленных лабораторий, реализующих преимущества цифровых измерительных комплексов на базе микроконтроллеров в объединении с облачными технологиями является одним из путей решения проблемы дистанционного лабораторного практикума по физике. В данной работе реализован новый принцип создания автоматизированного лабораторного практикума с удаленным доступом на примере лабораторной работы для определения скорости звука в воздухе методом стоячих волн в трубе.

Ключевые слова: дистанционный эксперимент, скорость звука, трубка Кундта, облачные технологии, микроконтроллер

1.Введение

В настоящее время наблюдается тенденция к активному применению технологий дистанционного образования в учебном процессе. Особенно актуальным это стало в связи пандемией, которая привела к онлайн обучению студентов вузах. В процессе решения этой задачи появилась проблема выполнения практических лабораторных работ онлайн. Вузы оказались не готовы к масштабному проведению таких работы и реальные эксперименты пришлось заменять на виртуальные. Не отрицая необходимость и таких учебных работ, все же они не должны заменять реальные эксперименты, поскольку имеют свою специфику. Кроме того, весьма важным является продвижение дистанционных технологий в лабораторные практикумы и в учебный эксперимент в целях повышения эффективности и снижения материальных затрат на обучение в сфере образования. Для этого создаются автоматизированные лабораторные практикумы с удаленным доступом (АЛП УД), которые позволяют проводить учебные эксперименты на реальном физическом оборудовании, находясь в любой точке планеты через глобальную сеть Интернет [1]. В современных условиях при использовании последних достижений в информационных и телекоммуникационных технологиях АЛП УД находят применение в учебном процессе, как в системе дистанционного и открытого образования [2, 3], так и при использовании традиционной очной технологии обучения. В данной работе была поставлена задача опробовать новый принцип функционирования лабораторных систем с возможностью удаленного доступа на основе лабораторной установки с удаленным доступом для определения скорости звука в воздухе методом стоячих волн в трубе [4].

Целью данной работы является создание автоматизированного лабораторного практикума по физике с удаленным доступом на базе микроконтроллера и облачного сервиса Firebase. При создании АЛП УД требуется: во-первых, применение специальных технических средств, как для автоматизации лабораторной установки, так и для связи управляющего компьютера или сервера с удалённым пользователем, проводящим эксперимент в режиме сетевого управления; во-вторых, разработка прикладного программного обеспечения или использование в отдельных случаях специализированных пакетов программ; в-третьих, методическая поддержка лабораторных учебно-научных экспериментов. Данная работа является продолжением цикла работ [5,6].

2. Дистанционное управление лабораторией на базе платформы Arduino через сеть

Разработка схемы удаленного доступа проводилась на основе выбора платформы Arduino с учетом ее особенностей. Общая схема удаленного доступа к экспериментальной установке, реализованной с помощью плат Arduino, в большинстве случаев включает в себя: плату общего назначения для сбора данных и управления устройствами, датчики и исполнительные механизмы экспериментальной установки, шилд для исполнительных механизмов экспериментальной установки, шилд или другую плату общего назначения для связи по локальной сети (LAN/WLAN), сервер, предоставляющий веб-доступ. Шилд – это плата расширения. Шилды можно разделить на полноразмерные и отдельные модули. Полноразмерные своими очертаниями повторяют форму платы Arduino, будь то UNO, Nano или MEGA. Отдельные модули – это платы произвольной формы, созданные для выполнения определенного набора функций. И те, и другие могут быть как общего назначения, так и для выполнения узконаправленных задач. Оснастив плату Arduino Ethernet или WiFi шилдом, можно запрограммировать её в простой веб-сервер, и с помощью доступа к этому серверу через браузер, запущенный на любом компьютере, который подключен к той же сети, что и Arduino, взаимодействовать с экспериментальной установкой. Такой веб-сервер, таким образом, заменяет управляющий экспериментальной установкой компьютер (его называют Labсервером). С другой стороны, плата с LAN/WLAN шилдом может взаимодействовать с существующим Lab-сервером, который также может предоставлять доступ к удаленному эксперименту. Рис. 1 иллюстрирует общую схему архитектуры удаленного доступа к экспериментальной установке, реализованных с помощью плат Arduino и других устройств, используемых в экспериментальной установке.

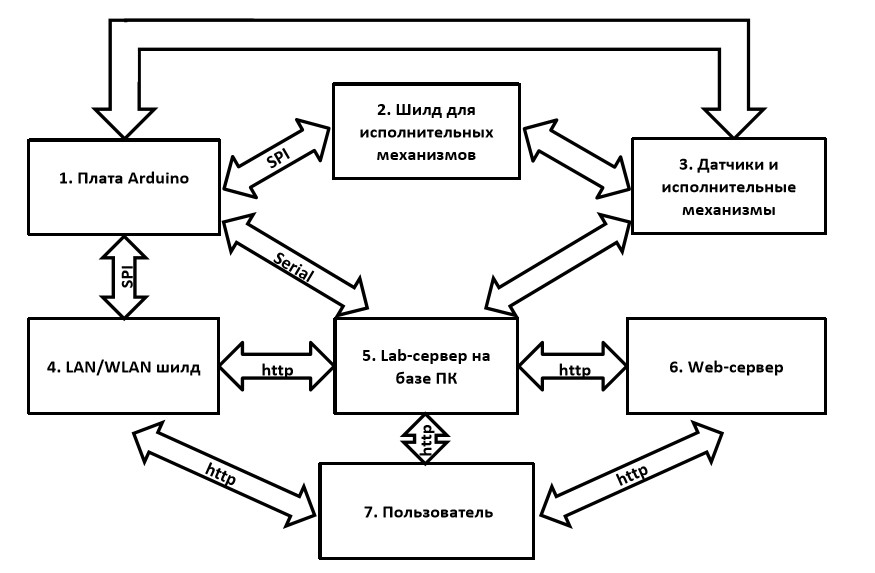


Рисунок 1. Общая схема удаленного доступа к лабораторной установке на базе платформы Arduino

Не все соединения на рис. 1 в действительности всегда требуются, так как это зависит от требований конкретного эксперимента. Центральной частью рис. 1 является плата Arduino общего назначения (1), аналоговые и цифровые входы которой подключены к датчикам экспериментальной установки (3) непосредственно проводами. Цифровые выходы могут быть подключены непосредственно или через соответствующий шилд (2) для исполнительных механизмов экспериментальной установки, если это необходимо. Плата общего назначения сообщается с шилдами главным образом через SPI (последовательный периферийный интерфейс). Веб-доступ для удаленного пользователя (7) может быть обеспечен различными способами. Как показано на рис. 1, Ethernet или Wifi шилд (4) для связи по локальной проводной или беспроводной сети, можно использовать, запрограммировав его как простой веб-сервер, который может быть непосредственно доступен удаленным пользователем. В сложных проектах лучшим решением является использование отдельного Lab-сервера с программным клиентом, а также использование простого веб-сервера, обеспечиваемого шилдом, только для передачи данных. В этом случае удаленный пользователь обращается к веб-серверу, размещенному на плате Arduino или на каком-либо существующем Lab-сервере на базе ПК (5). Затем запросы пользователей передаются программному клиенту, который запрашивает данные с платы Arduino (1) с помощью Ethernet или Wifi Shield (4) и направляет полученные данные через сервер обратно удаленным пользователям. Другой способ заключается в соединении (5) посредством проводной последовательной связи с платой Arduino (1), которая управляет экспериментом, обеспечивая к ней веб-доступ. Проводная последовательная связь между двумя платами может быть последовательной (UART) или TWI (двухпроводный интерфейс), он же I2C. Для управления экспериментом через глобальную сеть Интернет используется веб-сервер (6). Общая схема на рис.1 может быть при необходимости расширена другими соединениями двух или более плат Arduino или сетевыми соединениями в различных иерархических конфигурациях [7]. Есть еще несколько возможностей, которые не представлены на рис. 1. Во-первых, это использование платы общего назначения с интегрированным доступом к локальной сети, такой как HWAYEH UNO R3. Функционально это эквивалентно Arduino UNO R3 [8] с WiFi модулем ESP8266, но с несколько более низкой ценой. Во-вторых, вместо веб-сервера (6) для связи экспериментальной установки с удаленным пользователем по сети Интернет, можно использовать облачный сервис, например Firebase. В-третьих, как отмечалось, Ethernet или Wifi шилд, в случае HWAYEH UNO R3 – WiFi модуль ESP8266, можно использовать как простой веб-сервер и тем самым избавиться от Lab-сервера на базе ПК. Кроме того, модуль ESP8266 имеет 4МБ флеш-памяти, что освобождает нас от необходимости для хранения файлов веб-страниц использовать дополнительную память. С учетом этих возможностей, предполагая также, что никакие исполнительные механизмы в экспериментальной установке не используются, в данной работе была разработана новая схема удаленного доступа (рис. 2).

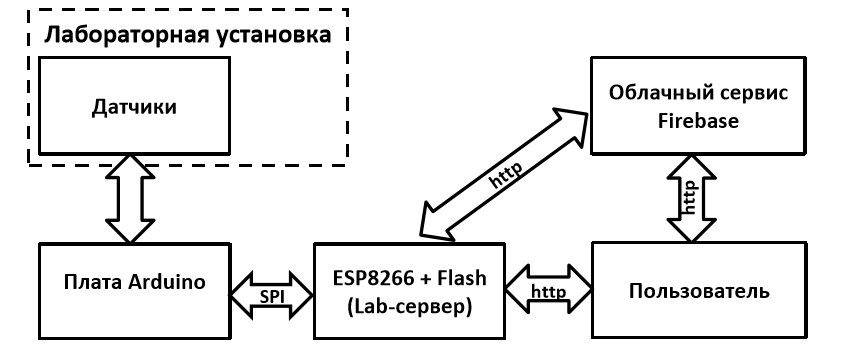


Рисунок 2. Схема удаленного доступа к лабораторной установке на базе платформы Arduino

Использование облачного сервиса Firebase аргументируется тем, что он упрощает задачу администрирования пользователей, предоставляет хостинг и доступ к базе данных реального времени. Также существуют библиотеки Arduino, позволяющие запрограммировать ESP8266 для работы с API данного сервиса напрямую [9].

Данная схема удаленного доступа была применена при создании лабораторной установки для определения скорости звука в воздухе методом стоячих волн в трубе [4].

3. Краткие теоретические сведения о распространении звука в замкнутой трубе

Для определения фазовой скорости распространения звуковой волны в воздухе используется связь этой величины с длиной волны и частотой звуковых колебаний [10, с. 265]:

 (1)

Звуковая волна, идущая от излучателя и волна, отраженная от другого конца трубы, ограниченного приемником звука (микрофоном), интерферирует в столбе воздуха, образуя стоячую волну. Такую трубу, открытую с обеих сторон, можно считать акустическим резонатором, в котором в виде устойчивых стоячих волн существуют только те, которые удовлетворяют условию [10, с. 285]:

(2)

Когда на длине трубы укладывается половина звуковой волны, что происходит при совпадении частоты внешнего воздействия динамика с основной частотой данной акустической системы, или кратное число половин, наблюдается увеличение акустического отклика системы в силу образования стоячих волн. Если плавно изменять частоту звуковых колебаний при неизменном значении длины трубы, будет наблюдаться последовательный ряд таких частот звуковых колебаний, при которых длина воздушного столба равна целому числу полуволн и воздушный столб в трубе звучит наиболее интенсивно.

Явление, при котором, наблюдается увеличение амплитуды колебаний системы при совпадении частоты внешнего воздействия с собственной частотой, называется резонансом. Определение скорости звука в настоящей работе основано на измерении разностей двух соседних частот, для которых имеет место усиление сигнала на приемнике звука. Согласно (1) и (2), соответствующая расчетная формула имеет вид

(3)

где c- фазовая скорость звука,L- длина трубы,  - среднее значение разностей соседних частот, на которых наблюдается резонансное усиление звука. Таким образом, определяя набор частот, на которых происходит усиление звуковых колебаний в некотором частотном диапазоне и, зная длину трубы, по формуле (3), можно определить скорость звука.

4. Аппаратное и программное обеспечение

4.1 Лабораторная установка

Лабораторная установка для определения скорости звука методом стоячих волн в трубе представляет собой горизонтальную пластиковую трубку, с одной стороны которой прикреплен пьезодинамик, а с другой —датчик звука (рис. 3).

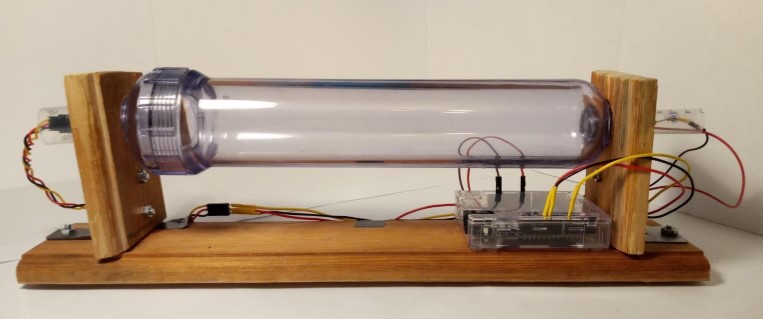


Рисунок 3. Лабораторная установка для определения скорости звука методом стоячих волн в трубе

Электрическая схема установки изображена на рис. 4.

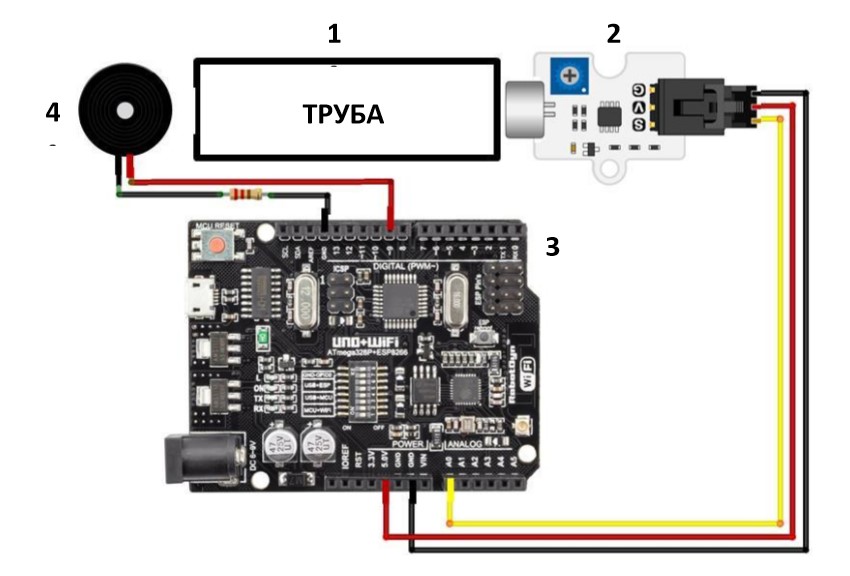


Рисунок 4. Электрическая схема лабораторной установки

Открытая с обоих концов труба (1) используется как акустический резонатор. Приемником звука служит модуль OCTOPUS EF04008 (2), размещенный у одного края трубы и соединенный с платой Arduino (3). Модуль подключается по схеме: вывод "V" модуля – к питанию +5 вольт, вывод "G" – к GND, вывод "S" – к аналоговому выводу "A0" платы Arduino. Пьезодинамик (4) размещен у другого края трубы и имеет два вывода. Положительный вывод подключен к цифровому выводу 9 платы Arduino с функцией ШИМ (PWM), в то время как отрицательный вывод заземлен на выводе GND Arduino через резистор на 100 Ом. Для создания лабораторной установки была использована печатная плата HWAYEH UNO R3 от китайского производителя. Это версия классической платы ARDUINO UNO R3. Полная интеграция микроконтроллера Atmel ATmega328 и IC Wi-Fi ESP8266 с флэш-памятью 4 МБ и конвертером USB-TTL CH340G на одной плате. Функционально это эквивалентно Arduino UNO R3 с Wi-Fi Shield и с SD Card Module.

4.2 Используемые программные средства

В данном разделе приводятся используемые в работе программные средства и их краткое описание. Программистская работа заключается в написании программ для модулей Arduino и разработке сайта для дистанционной работы с установкой. Программирование платы HWAYEH UNO R3 производится через последовательный интерфейс (Serial, UART) посредством встроенного конвертера USB-TTL через разъем USB Type C с использованием специальной среды разработки Arduino IDE на языке Wiring (упрощенная версия C++ для работы с микроконтроллерами) [11]. Для Arduino существует множество программных библиотек позволяющих реализовать самый различный функционал и взаимодействовать с внешней электроникой (датчики, сенсоры, моторы, реле, шины передачи данных и много другое). Получить дополнительную информацию по работе с платой можно на сайте производителя https://www.arduino.cc/en/Guide/ArduinoUno (на английском языке) или на множестве справочных сайтов и сайтов с примерами программ, например http://arduino.ru/Reference. В начале программы, как и в программах на C задаются переменные. Программа, написанная в среде Arduino, называется «скетч» и состоит из двух функций, setup() и loop(). setup() — функция setup() автоматически выполняется в самом начале программы Arduino. Внутри этой функции инициализируют переменные, пины (выводы платы) и любые библиотеки, которые используются в программе. Функция setup() запускается только один раз во время выполнения скетча Arduino, прямо при включении питания или сбросе.

loop() — функция loop() содержит основную часть программы. Эта функция выполняется после завершения setup(). Arduino будет выполнять команды внутри цикла снова и снова, пока плата не выключится. После написания программы ее необходимо скомпилировать и загрузить на устройство с помощью среды разработки Arduino. После успешной загрузки устройство нужно перезагрузить. Разработка сайта осуществлялась в редакторе кода Visual Studio Code с помощью технологий HTML, CSS и JavaScript специально для выбранной лаборатории удалённого доступа. Также используется облачный сервис Firebase, который предоставляет базу данных реального времени и позволяет обходиться без фактического написания серверной части.

4.3 База данных Firebase Realtime

База данных Firebase Realtime - это облачная NoSQL база данных компании Firebase. Данные хранятся в формате JSON и синхронизируются в режиме реального времени с каждым подключенным клиентом. Поддержаны особенности интеграции с приложениями под операционные системы Android и iOS, реализовано API для приложений на JavaScript, Java, Objective-C и Node.js, также возможно работать напрямую с базой данных в стиле REST из ряда JavaScript-фреймворков, включая AngularJS, React, Vue.js, Ember.js и Backbone.js. Предусмотрено API для шифрования данных. При создании кроссплатформенных приложения с помощью SDK для iOS, Android и JavaScript, все клиенты совместно используют один экземпляр базы данных в реальном времени и автоматически получают обновления с самыми новыми данными [12]. Ключевые особенности Firebase Realtime:

Работа в реальном времени. Вместо типичных HTTP-запросов база данных Firebase Realtime использует синхронизацию данных - каждый раз, когда данные меняются, любое подключенное устройство получает это обновление в течение миллисекунд,что позволяет обеспечивать совместный и захватывающий опыт, не думая о сетевом коде. Оффлайн работа. Приложения Firebase остаются отзывчивыми даже в автономном режиме, поскольку SDK базы данных Firebase Realtime сохраняет данные пользователя на диск. После восстановления соединения клиентское устройство получает любые пропущенные изменения, синхронизируя его с текущим состоянием сервера.

Доступность с устройств клиентов.

Доступ к базе данных Firebase Realtime можно получить непосредственно с мобильного устройства или веб-браузера; нет необходимости в сервере приложений. Безопасность и проверка данных доступны через правила безопасности баз данных Firebase Realtime, основанные на выражениях правила, которые выполняются при чтении или записи данных.

Масштабирование по нескольким базам данных.

С помощью базы данных Firebase Realtime можно масштабировать потребности приложений в данных, разбивая данные на несколько экземпляров базы данных в одном проекте Firebase. Есть возможность оптимизировать аутентификацию с помощью Firebase Authentication для конкретного проекта и проверять подлинность пользователей во всех экземплярах базы данных. Управлять доступом к данным в каждой базе данных с помощью пользовательских правил базы данных Firebase Realtime для каждого экземпляра базы данных. База данных Firebase Realtime позволяет создавать многофункциональные приложения для совместной работы, обеспечивая безопасный доступ к базе данных непосредственно из кода на стороне клиента. Данные сохраняются локально, и даже в автономном режиме события в реальном времени продолжают срабатывать, предоставляя пользователю полноценный отклик. Когда устройство восстанавливает соединение, база данных реального времени синхронизирует локальные изменения данных с удаленными обновлениями, которые произошли, когда клиент находился в автономном режиме, автоматически объединяя любые изменения. База данных реального времени предоставляет гибкий язык правил на основе выражений, называемый правилами безопасности баз данных в реальном времени Firebase, для определения того, как данные должны быть структурированы и когда данные могут считываться или записываться. При интеграции с Аутентификацией Firebase разработчики могут определить, кто имеет доступ к каким данным и как они могут получить к ним доступ. База данных реального времени является базой данных NoSQL и, поэтому имеет иные виды оптимизации и функциональность по сравнению с реляционной базой данных. API-интерфейс базы данных реального времени позволяет выполнять только те операции, которые могут быть выполнены быстро. Это позволяет производить обработку данных в реальном времени, обслуживая миллионы пользователей без ущерба для скорости отклика. В связи с этим важно продумать то, как пользователи должны получать доступ к данным, и затем соответствующим образом структурировать их [13].

4.4 Описание практической части

4.4.1 Программирование Arduino

Для того чтобы разобраться в особенностях программирования используемой платы HWAYEH UNO R3 рассмотрим ее структурную схему (рис.5)

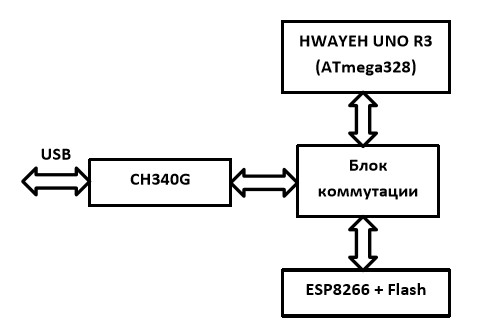


Рисунок 5. Структурная схема платы HWAYEH UNO R3

На одной плате интегрированы микропроцессорные модули Atmel ATmega328 и IC Wi-Fi ESP8266 с флэш-памятью 4 МБ и конвертер USB-TTL CH340G. Все модули могут работать вместе или независимо. Блок коммутации представляет собой набор из 8 DIP переключателей и чип USB порта CH340G, который может поочередно подключается к каждому из модулей. Схематическое представление возможных режимов работы платы изображено на рис. 6.

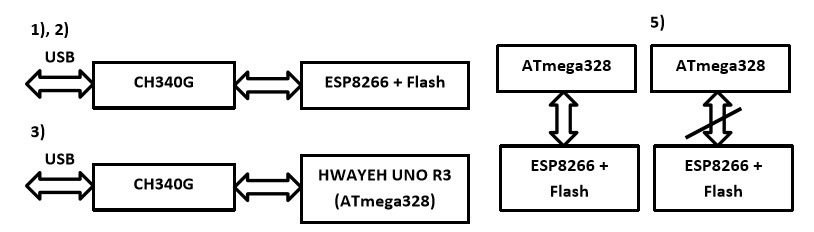


Рисунок 6. Схематическое представление режимов работы HWAYEH UNO R3

Каждому режиму работы платы соответствует определенное положение переключателей . Так, например, для того чтобы запрограммировать модуль ESP8266, необходимо выставить положения переключателей 5, 6 и 7 в положение «ON», а 1, 2, 3, 4 и 8 в положение «OFF». Аналогично делается для модуля ATmega328, только в этом случае в положение «ON» нужно выставить только 1-й и 2-й переключатели. Для того чтобы модули начали работать вместе необходимо в очередной раз изменить положения переключателей и выставить их таким образом, чтобы два чипа были соединены между собой. При этом передача данных между модулями происходит через последовательный порт (также известный как UART или USART): Serial.

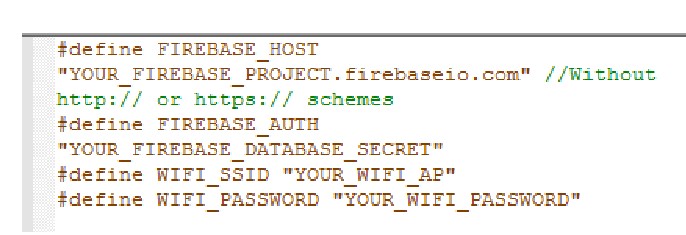
Особенности использования данной платы: 1. Наличие двух микроконтроллеров, необходимость создания скетчей для каждого. 2. Для совместной работы модулей необходимо разработать протокол обмена данными между модулями. 3. Для программирования модулей необходимо каждый раз менять положение переключателей. 4. В режиме совместной работы модулей нет возможности через USB порт выводить отладочную информацию. Работа по программированию модулей Arduino заключается в написании кода для основного модуля платы (ATmega328) и кода для WiFi модуля ESP8266.

4.4.2 Написание кода для основного модуля платы

В данном скетче при получении численного значения частоты сигнала на порт основного модуля генерируется сигнал этой частоты, вычисляется относительная амплитуда сигнала и передается обратно. В функции setup() с помощью функции Serial.begin() инициируем последовательное соединение и задаем скорость передачи данных 9600 бит/c (бод). В функции loop() с помощью конструкции if и условия Serial.available() проверяется наличие байт (символов) доступных для считывания из буфера последовательного порта. (Под символами понимаются данные, которые уже приняты и хранятся в последовательном приемном буфере, который может хранить максимум 64 байта). Далее, с помощью функции Serial.readString() считываем данные из буфера последовательного порта в строку, и, в зависимости от принятого значения, либо меняем частоту сигнала, генерируемого пьезодинамиком, либо подаем на вывод прямоугольный сигнал заданной частоты (с коэффициентом заполнения 50%) – функция tone(). С помощью функции analogRead() считываем значение с указанного аналогового входа. Большинство плат Arduino имеют 6 каналов (8 каналов у платы Mini и Nano, 16 у Mega) c 10-битным аналого-цифровым преобразователем (АЦП). Напряжение, поданное на аналоговый вход, обычно от 0 до 5 В будет преобразовано в значение от 0 до 1023, это 1024 шага с разрешением 0,0049 В. Считывание значения с аналогового входа занимает примерно 100 микросекунд (0,0001 с), т.е. максимальная частота считывания приблизительно 10000 раз в секунду. Ищем максимальное и минимальное значение амплитуды и с помощью Serial.println(). Вызываем функцию noTone(), которая прекращает генерирование прямоугольного сигнала после использования функции tone(). На порт посылаем строку символов – значение размаха амплитуды сигнала. После написания программы ее необходимо скомпилировать и загрузить на устройство с помощью среды разработки Arduino IDE, выбрав в настройках плату «Arduino Uno». Далее, необходимо изменить положение DIP переключателей на плате, задав соответствующий режим работы. После успешной загрузки, устройство нужно перезагрузить.

4.4.3 Написание кода для WiFi модуля ESP8266

Модуль ESP8266 программируется как простой веб-сервер, включающий обработчик событий Firebase. Для работы с API сервиса Firebase была использована Arduino библиотека Firebase-ESP8266 [14]. Кроме того, в скетче используются библиотеки: ESP8266WiFi, WiFiClient, ESP8266WebServer. Примеры скетчей для каждой из библиотек можно посмотреть, выбрав нужную библиотеку в пункте меню Файл | Примеры в Arduino IDE или на сайте разработчика. В начале программы необходимо инициализировать переменные для установки связи по WiFi и с облачным сервисом Firebase (Листинг 1).



Листинг 1. Задание переменных проекта

Значение переменных WIFI\_SSID и WIFI\_PASSWORD задаются в зависимости от локальной сети, в которой работает лабораторная установка. Значения переменных FIREBASE\_HOST и FIREBASE\_AUTH неизменны и зависят от конкретного проекта Firebase. Их можно посмотреть в консоли разработчика Firebase. Для работы по локальной сети скетч содержит следующие функции:

is\_authenticated() – проверка авторизации через Cookie.

handleLogin() – для проверки введенных пользователем авторизационных данных.

handleRoot() – функция для отправки страницы лаборатории на запрос пользователя. handleNotFound() – функция для ответа на запрос несуществующей страницы.

getStrength() – функция для информирования пользователя о вводе неверных авторизационных данных. В функции setup() осуществляется настройка сети WiFi и связь с облачным сервисом Firebase. В функции loop() запускается веб-сервер и обработчики событий базы данных Firebase. Полный код скетча находится в GitHub репозитории по адресу: https://github.com/CodeImm/ArduinoLab. После написания программы ее необходимо скомпилировать и загрузить на устройство с помощью среды разработки Arduino IDE, выбрав в настройках плату «Generic ESP8266 Module» и предварительно изменив положение DIP переключателей на плате, задав соответствующий режим работы. После успешной загрузки, устройство нужно перезагрузить. Т. к. программный код скетча выполняется последовательно и зациклен, работать с установкой можно через локальную сеть и облачный сервис одновременно. Однако, ввиду того, что сбор и передача данных сама по себе занимает некоторое время, есть некоторые ограничения по одновременному обращению к установке нескольких пользователей. Наиболее оптимальным представляется обеспечение доступа одному пользователю по сети интернет, и одному-двум пользователям по локальной сети или наоборот (большее количество пользователей может вызывать значительные задержки ответов по времени при последовательной обработке запросов).

Выводы по работе с платой HWAYEH UNO R3:

1. Очень удобная техническая реализация беспроводной связи.

2. Отладка и процесс написания кода очень неудобный и требует большого количества манипуляций с DIP переключателями.

3. Альтернативой данной платы может быть комплект из двух различных плат, что значительно менее удобно и дороже.

4.4.4 Разработка сайта

Для работы пользователя с установкой по сети Интернет в данной работе вместо веб-сервера используется облачный сервис Firebase. Взаимодействие пользователя с лабораторной установкой осуществляется через веб-интерфейс. Веб-интерфейс для функционирования АЛП УД должен, с одной стороны, обслуживать в интерактивном режиме диалог удалённого пользователя с облачным сервисом при настройке условий эксперимента, а с другой — реализовать заданный режим на лабораторной установке и трансляцию результатов его выполнения. Кроме того, разрабатываемый для данной АЛП УД сайт должен обеспечивать методическую поддержку лабораторной работы, т.е. содержать описание лабораторного стенда, методику измерений, различные справочные материалы и т.п. Созданный сайт АЛП УД включает в себя систему авторизации пользователей, реализованную с помощью встроенных служб Firebase, методическую поддержку лабораторной работы в виде отдельной страницы с методическим пособием и страницу с веб-интерфейсом для работы с лабораторной установкой и возможностью сохранять результаты работы. Загрузка сайта осуществляется с помощью стандартного Интернет обозревателя по протоколу HTTP. Взаимодействие с лабораторной установкой осуществляется через NoSQL базу данных (БД) реального времени Firebase с помощью ее слушателей. Асинхронный слушатель базы данных сообщает пользователю о хранящихся данных. Этот привязанный к списку ожидающих уведомлений слушатель активируется один раз в начале для первичного набора данных и потом срабатывает каждый раз при изменении данных. Рассмотрим пример работы программного кода при снятии измерений с лабораторной установки (листинг – 2).



Листинг 2. Обработчик событий, срабатывающий при нажатии на кнопку генерации сигнала

Когда пользователь вводит в специальное поле формы частоту сигнала и нажимает на кнопку генерации (HTML-элемент «addData»), срабатывает обработчик событий, в котором, при условии, что введенное пользователем значение частоты валидно (функция valid()), эта кнопка блокируется (функции prop() и addClass(). Далее, объявляется переменная «frequency», которой присваивается введенное значение из поля, объявляется переменная ссылки на узел БД «status», в котором хранятся атрибуты, характеризующие состояние установки (рис. 7).

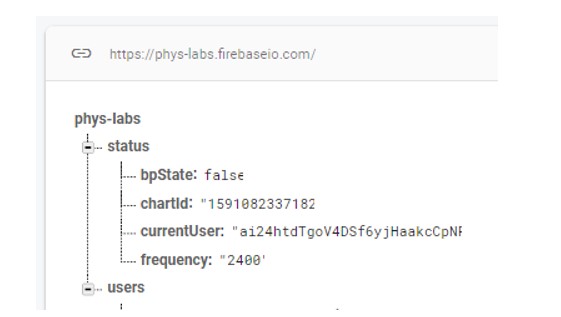


Рисунок 7. Информация о состоянии лабораторной установки в базе данных Firebase

После чего, вызывается функция update() для обновления значения узла «frequency» на введенную частоту и «bpState» с false на true, что говорит о том, что необходимо сгенерировать сигнал на удаленной лабораторной установке. На установке выполняется слушатель узла «bpState», считывается значение частоты, генерируется сигнал, значение «bpState» в БД меняется на false и в БД записываются результаты измерения. После того, как в узел БД записались результаты измерения, на сайте автоматически срабатывает слушатель, привязанный к этому узлу. Полученные данные разбираются и отображаются на странице. Кнопка генерации сигнала снова становиться активной (листинг 3).



Листинг 3. Слушатель узла БД, в который записываются результаты измерений

Полный код проекта находится в GitHub репозитории по адресу: <https://github.com/CodeImm/ArduinoLabWebSite>.

В результате работы были написаны и протестированы программы для модулей Arduino, входящих в лабораторную установку. Также разработан сайт для работы с установкой, который реализует следующий функционал: регистрация на сервисе; вход под своей учетной записью на сервис; методическая поддержка лабораторной работы; управление удаленной лабораторной установкой; сохранение результатов работы.

4.5 МЕТОДИКА ИСПОЛЬЗОВАНИЯ ЛАБОРАТОРИИ

4.5.1 Общие положения

При использовании АЛП УД через сеть Интернет предполагается, что учащийся перед проведением удалённого эксперимента принимал участие в более простых лабораторных работах, проводимых в своём образовательном учреждении, и имеет элементарные понятия о теме проводимого практикума, а также необходимые практические навыки для его проведения [15, 16, 17]. Для проведения лабораторного практикума учащиеся направляют заявки на выполнение лабораторной работы преподавателю. Чаще всего переписка для согласования времени проведения работы ведётся по электронной почте, но могут быть проведены и прямые аудиовизуальные переговоры с диспетчером (или системным программистом) объекта исследования, например, с использованием сред Zoom или Skype. Сайт практикума находиться по адресу https://phys-labs.web.app/. Взаимодействие с установкой осуществляется с помощью веб-интерфейса через базу данных реального времени. Для работы с установкой сервер запрашивает авторизационные данные и регистрацию на сайте.

4.5.2 Взаимодействие с установкой

Для входа на сервер АЛП УД, учащемуся необходимо зарегистрировать свою учетную запись (рис. 8).

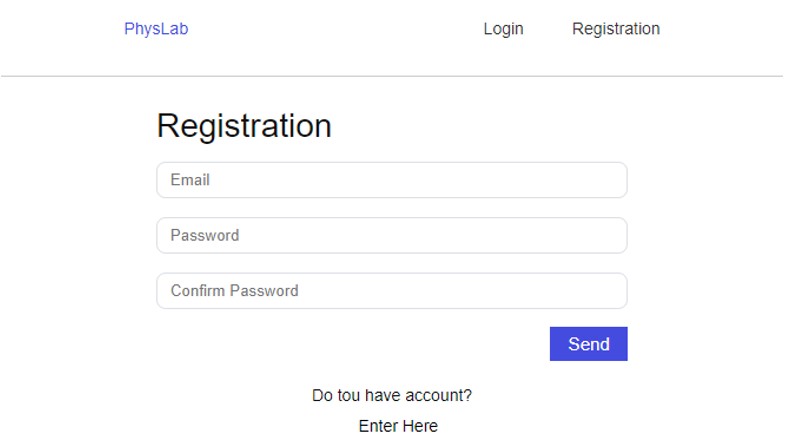


Рисунок 8. Страница регистрации

После входа по созданной учетной записи, учащийся переходит на страницу с методическим пособием (рис. 9) для ознакомления с теоретической частью, описанием оборудования и методикой проведения эксперимента.

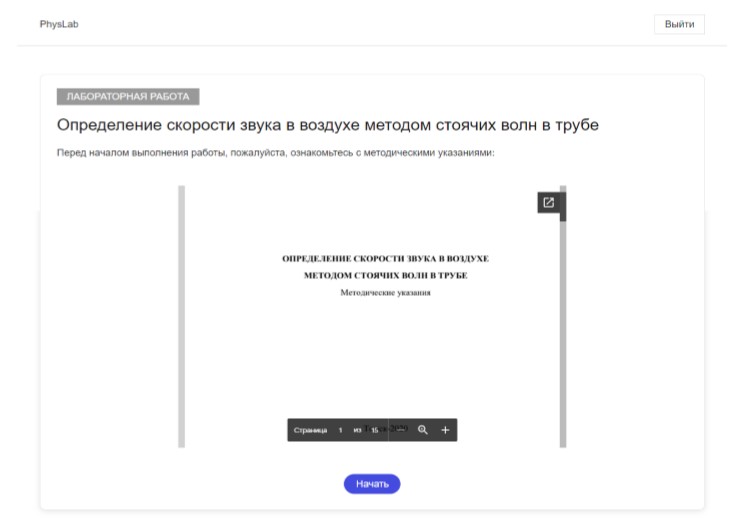


Рисунок 9. Страница с методическим пособием

Далее типовой сценарий проведения удалённого учебного эксперимента выглядит следующим образом. Учащийся нажимает на кнопку «Начать», осуществляя вход в монопольный режим управления удаленной лабораторной установкой, переходит на страницу с лабораторной работой (рис. 10), содержащую панель управления.

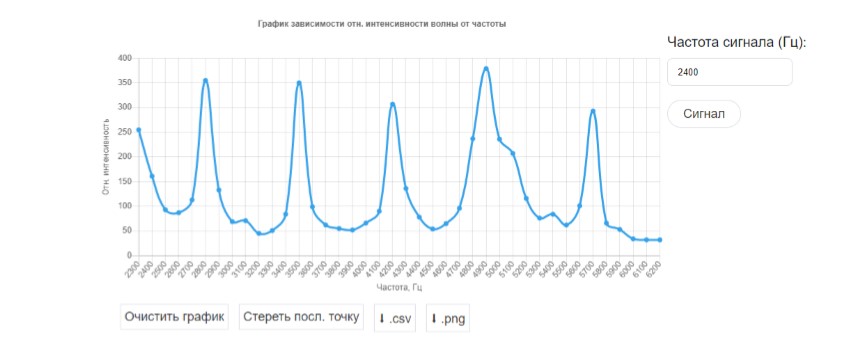


Рисунок 10. Веб-интерфейс лабораторной работы с графиком

Слева на странице находится рабочее поле, справа – поле ввода частоты сигнала и кнопка генерации сигнала. Внизу расположены кнопки для очистки графика, удаления последней отмеченной точки и кнопки для сохранения результатов работы в форматах CSV и PNG.

По окончании работы учащийся сохраняет полученные им результаты в табличном и/или графическом виде для обработки полученных данных и выходит из монопольного режима управления. Далее составляет отчет, сохраняет его в одном из форматов для хранения электронных документов и отправляет его преподавателю для проверки.

5.Заключение

В результате работы на основе обзора существующих решений построения лабораторных систем был создан бюджетный автоматизированный лабораторный практикум с удаленным доступом на базе платформы Arduino. Практикум включает в себя одну лабораторную работу для определения скорости звука в воздухе методом стоячих волн в трубе. Была разработана схема удаленного доступа к лабораторной установке с использованием облачных технологий, требующая минимального количества ресурсов. Были написаны и протестированы программы для модулей Arduino, входящих в лабораторную установку, разработан сайт для работы с установкой удаленно. Созданный практикум на базе платформы Arduino был успешно апробирован в образовательном процессе на кафедре общей и экспериментальной физики ТГУ. Использование практикума возможно не только в рамках дистанционного и открытого образования, но и при проведении традиционных очных учебных занятий.

ЛИТЕРАТУРА

1. Журавлева О.Б., Крук Б.И. Соломина Е.Г. Управление Интернет обучением в высшей школе// под ред. Б.И. Крука. – Новосибирск: «Веди», 2005. –255 с.

2. Норенков И.П. Системные вопросы дистанционного обучения // Информационные технологии. – 2001. - №3. – С.17-21.

3. Образование и XXI век: Информационные и коммуникационные технологии / Под ред. В.Г. Кинелева. – М.: Наука, 1999. – 191 с.

4. Горчаков Л.В., Колесников Д.А., Печерицин А.А. Акустический интерферометр Квинке и его использование для определения скорости звука//Проблемы учебного физического эксперимента. – Вып.31.-М.-ИСРО РАО, – 2020. – С.54-56.

5.Горчаков Л.В.,Малых Д.О. Дистанционная лабораторная работа «Изучение зависимости емкости переменного конденсатора от угла поворота» //Проблемы учебного физического эксперимента. – Вып.29.-М.-ИСРО РАО, – 2019. – С.71-72.

6. Горчаков Л.В. Определение скорости звука и других физических параметров на основе трубки Кундта//Проблемы учебного физического эксперимента. – Вып.32.-М.-ИСРО РАО, – 2020. – С.88-89.

7. V.M. Cvjetkovi, M.Matijevi // Overview of architectures with Arduino boards as building blocks for data acquisition and control systems, 13 th International Conference on Remote Engineering and Virtual Instrumentation (REV), – 2016. [Электронный ресурс]. URL: https://doi.org/10.3991/ijoe.v12i07.5818 (дата обращения: 10.03.2020).

8. Официальный сайт Arduino UNO [Электронный ресурс]. URL: https://www.arduino.cc/en/Guide/ArduinoUno (дата обращения: 10.03.2020).

9. Firebase-ESP8266 GitHub: [Электронный ресурс]. URL: https://github.com/mobizt/Firebase-ESP8266 (дата обращения: 10.03.2020).

10. Савельев И.В. Курс общей физики : Т. 1. – М. : Наука, 1970. – 517 с.

11. Петин В.А. Проекты с использованием контроллера Arduino. 2-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2015.

12. Firebase RealTime Database // Firebase Guides. [Электронный ресурс]. URL: https://firebase.google.com/docs/database?hl=ru (дата обращения: 11.05.2020)

13. Introduction // Firebase Guides. [Электронный ресурс]. URL: https://firebase.google.com/docs/database (дата обращения: 11.05.2020)

14. Firebase-ESP8266 GitHub: [Электронный ресурс]. URL: https://github.com/mobizt/Firebase-ESP8266 (дата обращения: 10.03.2020).

15. Зимин А.М. Лаборатории удаленного доступа в практической подготовке инженеров XXI века. М.: Изд-во МГТУ им. Н.Э. Баумана, 2007

16. Лабораторный практикум с удалённым доступом как средство практической подготовки специалистов в техническом университете // И.Б. Федоров, А.М. Зимин, С.В. Коршунов, Ю.М. Кузнецов / Индустрия образования/ Под ред. А.А. Полякова, Ю.М. Кузнецова, Г.Ф. Филаретова, М.Б. Дружининой. М.: МГИУ, 2002. Вып. 2. С. 312–319

17. Зимин А.М., Маслов С.И. Практическая подготовка специалистов на основе технологии удалённого доступа к экспериментальным стендам через глобальную сеть // Открытое образование. 2009. № 5. С. 94–100.