МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вологодский государственный университет»**

**Институт математики, естественных и компьютерных наук**

**Информатика и вычислительная техника**

**КУРСОВОЙ ПРОЕКТ**

Детектор шума.

Дисциплина: «Микропроцессорные системы»

Направление подготовки: 09.03.01. Информатика и вычислительная техника

|  |  |
| --- | --- |
| Руководитель | Давыдова Е.Н. |
| Выполнили студенты | Пчелкина О.С. |
| Группа, курс | ВМ-31 |
| Дата сдачи | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Дата защиты | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись преподавателя)* |

Вологда

2022 г.

ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 2](#_Toc105335677)

[ВВЕДЕНИЕ 3](#_Toc105335678)

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#_Toc105335679)

[1.1 Применение устройства 4](#_Toc105335680)

[1.2 Использование плат и среды программирования arduino 6](#_Toc105335681)

[1.3 Использование микроконтроллеров 7](#_Toc105335682)

[1.4 Классификация и структура микроконтроллеров 8](#_Toc105335683)

[1.5 Структура процессорного ядра микроконтроллера 10](#_Toc105335684)

[1.6 Принцип действия детектора шума 13](#_Toc105335685)

[2. ПРАКТИЧЕСКАЯ ЧАСТЬ 14](#_Toc105335686)

[2.1 Необходимые платы для создания устройства 14](#_Toc105335687)

[2.2 Подключение элементов 16](#_Toc105335688)

[2.3 Программная составляющая 18](#_Toc105335689)

[ЗАКЛЮЧЕНИЕ 19](#_Toc105335690)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 20](#_Toc105335691)

ВВЕДЕНИЕ

Шумом называются беспорядочные колебания различной физической природы, отличающиеся сложностью временных и спектральных характеристик. Шум - один из факторов физического загрязнения окружающей среды. Интенсивное шумовое воздействие вызывает в органе слуха изменения, составляющие специфическую реакцию организма.

Исследования ученых показали, что шумовое загрязнение окружающей среды является одним из самых опасных факторов, влияющих на здоровье человека. Под этой формой загрязнения понимают превышение естественного уровня шума и ненормальное изменение звуковых характеристик на рабочих местах, в населенных пунктах и других местах. Источниками шума являются работа транспорта, шумы, создаваемые самолетами, работа промышленных устройств, бытовых приборов и другие причины. Раздражающее действие звука на человека зависит от его интенсивности, спектра и продолжительности воздействия.

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Шумомер — прибор для объективного измерения уровня звука в производственных или бытовых условиях.

* 1. Применение устройства

Шумомер измеряет уровень посторонних звуков, шума. Его используют, чтобы определять:

1. громкость звучания, издаваемого акустическими системами;
2. соответствие рабочих мест нормам при их аттестации;
3. уровень шума в любых помещениях, на стройплощадках, производственных объектах;
4. качество защищающих от посторонних звуков средств во время их тестирования.

Измерители уровня шума позволяют контролировать эффективность звукоизоляционных систем, установленных в помещениях. В строительной сфере ими пользуются с целью проверки соответствия звукоизоляционных стройматериалов тем характеристикам, которые заявляют их производители. Оценивается уровень изоляции строительных объектов от постороннего шума.

Установлены нормативные показатели для шума, которые должны соблюдаться днем и ночью в жилом помещении.

Шумомеры бывают:

Простыми – аппаратами, выполняющими однократные операции, при которых отображаются значения во время их активации. Определяют звуковую громкость в децибелах.

Комплексными – представляющими собой более сложное оборудование. С их помощью проводят серии измерений, что позволяет анализировать путем составления графиков звуковых колебаний, оценивать полученные за определенный период результаты. Применяются на производственных объектах.

Измерители уровня шума бывают разной точности и в зависимости от этого подразделяются на следующие классы:

0 – представителей данной группы считаются эталонами, поскольку они отображают максимально точные результаты. Подобное устройство дорогое и не часто применяется.

1 – в сравнении с предыдущим оборудованием проще, но все же считается очень точным. Зачастую им пользуются в условиях лабораторий, но также применяют и в других ситуациях, когда возникает необходимость высокоточного измерения уровня звука.

2 – этой группой приборов наиболее часто пользуются в производственных условиях, в сфере строительства.

3 – группа устройств, имеющих простейшую конструкцию. Такие модели недорогие, они способны решать бытовые задачи по измерению шума. Поскольку отображают ориентировочные значения показателей, то для профессионального использования не подходят.

* 1. Использование плат и среды программирования arduino

Аrduino – это программно-аппаратная платформа для сверхбыстрого создания электронных устройств, поддерживаемая разработчиками по всему миру и пользующаяся огромной популярностью в России и за рубежом.

Платформа позволяет общаться и взаимодействовать с окружающим миром с помощью всевозможных датчиков, сенсоров, моторов и других узлов.

Платформа состоит из двух частей: программной и аппаратной.  
В качестве программной части выступает кроссплатформенная среда разработки Arduino IDE, которая может запускаться на операционных системах Windows, Linux, Mac OS. С помощью данной среды можно писать код и программировать платы. В комплекте с программой поставляются многочисленные примеры, библиотеки и удобные утилиты. В качестве аппаратной части выступают различные платы. На сайте производителя их насчитывается уже более двух десятков разновидностей. Так как платформа распространяется весьма свободно, то в продаже можно найти множество клонов и различных плат расширения.

Рассмотрим основные преимущества данной платформы.

* Кроссплатформенная среда разработки. В отличие от многих сред программирования, Arduino не ограничивает свободу выбора операционной системы.
* Отсутствие необходимости в программаторе. Почти все платы имеют USB разъем. Для плат, в которых не предусмотрены USB, продаются дешевые переходники для подключения USB.
* Наличие большого числа плат. Существует несколько десятков видов оригинальных плат для разных задач, а также их многочисленные клоны.
* Переносимость кода. Написав один раз код для платы Arduino UNO, вы можете перенести его на более мощную плату Arduino MEGA или более слабую Arduino NANO. Никаких исправлений в коде делать не придется.
* Отсутствие необходимости в пайке. Схемы собираются на беспаячной макетной плате с помощью специальных проводков.
* Язык программирования C/C++ – один из самых популярных языков программирования.
* Наличие большого числа плат-расширений. С ними платы Arduino превращаются в конструктор. Можно добавить сетевую плату Ethernet, плату Bluetooth, GPS, GSM и даже видеоплату VGA.

Среда разработки Arduino состоит из встроенного текстового редактора программного кода, области сообщений, окна вывода текста(консоли), панели инструментов с кнопками часто используемых команд и нескольких меню. Для загрузки программ и связи среда разработки подключается к аппаратной части Arduino.

Программа, написанная в среде Arduino, называется скетч. Скетч пишется в текстовом редакторе, имеющем инструменты вырезки/вставки, поиска/замены текста. Во время сохранения и экспорта проекта в области сообщений появляются пояснения, также могут отображаться возникшие ошибки. Окно вывода текста(консоль) показывает сообщения Arduino, включающие полные отчеты об ошибках и другую информацию.

* 1. Использование микроконтроллеров

Микроконтроллер – это «мозг» Ардуино, он выполняет программу, хранит промежуточные результаты в своей памяти, совершает все логические и арифметические операции, дает команды портам ввода/вывода. Внутри этого мозга есть хранилище, в котором находится программа. Будучи один раз туда записанной, она может храниться там десяток лет. Объем этой программы не может превышать 30 килобайт. Это примерно соответствует 30 тысячам элементарных команд микроконтроллера.

Программа может перезаписываться не менее 1000 раз. Кварцевый резонатор заставляет работать МК на определенной частоте, без него микроконтроллер просто не сможет работать. По сути, это «сердце» Ардуино, бьющееся 16 миллионов раз в секунду. И за каждый удар такого сердца микроконтроллер выполняет одну команду из его программы. Максимальная частота, на которой может работать микроконтроллер – 20 мегагерц, т.е. 20 миллионов команд в одну секунду. Однако следует обратить внимание, что это не те команды, которые мы будем писать для программы. Одна написанная нами команда может превратиться в несколько десятков или даже несколько тысяч команд самого микроконтроллера. Но они всё равно будут очень быстро выполняться. Как правило, микроконтроллеры на платах Ардуино работают с частотами 16 или 8 мегагерц. Чем больше частота, тембольше потребление энергии микроконтроллером.

* 1. Классификация и структура микроконтроллеров

В настоящее время выпускается целый ряд типов МК. Все эти приборы можно условно разделить на три основных класса:

* 8-разрядные МК для встраиваемых приложений;
* 16 - и 32-разрядные МК;
* цифровые сигнальные процессоры (DSP).

Наиболее распространенным представителем семейства МК являются 8-разрядные приборы, широко используемые в промышленности, бытовой и компьютерной технике. Они прошли в своем развитии путь от простейших приборов с относительно слаборазвитой периферией до современных многофункциональных контроллеров, обеспечивающих реализацию сложных алгоритмов управления в реальном масштабе времени. Причиной жизнеспособности 8-разрядных МК является использование их для управления реальными объектами, где применяются, в основном, алгоритмы с преобладанием логических операций, скорость обработки которых практически не зависит от разрядности процессора.

Росту популярности 8-разрядных МК способствует постоянное расширение номенклатуры изделий. Современные 8-разрядные МК обладают, как правило, рядом отличительных признаков. Перечислим основные из них:

* модульная организация, при которой на базе одного процессорного ядра (центрального процессора) проектируется ряд (линейка) МК, различающихся объемом и типом памяти программ, объемом памяти данных, набором периферийных модулей, частотой синхронизации;
* использование закрытой архитектуры МК, которая характеризуется отсутствием линий магистралей адреса и данных на выводах корпуса МК. Таким образом, МК представляет собой законченную систему обработки данных, наращивание возможностей которой с использованием параллельных магистралей адреса и данных не предполагается;
* использование типовых функциональных периферийных модулей (таймеры, процессоры событий, контроллеры последовательных интерфейсов, аналого-цифровые преобразователи и др.), имеющих незначительные отличия в алгоритмах работы в МК различных производителей;
* расширение числа режимов работы периферийных модулей, которые задаются в процессе инициализации регистров специальных функций МК.

При модульном принципе построения все МК одного семейства содержат процессорное ядро, одинаковое для всех МК данного семейства, и изменяемый функциональный блок, который отличает МК разных моделей. Процессорное ядро включает в себя:

* центральный процессор;
* внутреннюю контроллерную магистраль (ВКМ) в составе шин адреса, данных и управления;
* схему синхронизации МК;
* схему управления режимами работы МК, включая поддержку режимов пониженного энергопотребления, начального запуска (сброса) и т.д.

Изменяемый функциональный блок включает в себя модули памяти различного типа и объема, порты ввода/вывода, модули тактовых генераторов (Г), таймеры. В относительно простых МК модуль обработки прерываний входит в состав процессорного ядра. В более сложных МК он представляет собой отдельный модуль с развитыми важными возможностями. В состав изменяемого функционального блока могут входить и такие дополнительные модули как компараторы напряжения, аналого-цифровые преобразователи (АЦП) и другие. Каждый модуль проектируется для работы в составе МК с учетом протокола ВКМ. Данный подход позволяет создавать разнообразные по структуре МК в пределах одного семейства.

* 1. Структура процессорного ядра микроконтроллера

Самыми главными и определяющими характеристиками, производительности процессорного ядра МК, являются:

* набор регистров для хранения промежуточных данных;
* система команд процессора;
* способы адресации операндов в пространстве памяти;
* организация процессов выборки и исполнения команды.

С точки зрения системы команд и способов адресации операндов процессорное ядро современных 8-разрядных МК реализует один из двух принципов построения процессоров:

* процессоры с CISC-архитектурой, реализующие так называемую полную систему команд (Complicated Instruction Set Computer);
* процессоры с RISC-архитектурой, реализующие сокращенную систему команд (Reduced Instruction Set Computer).

CISC-процессоры выполняют большой набор команд с развитыми возможностями адресации, давая разработчику возможность выбрать наиболее подходящую команду для выполнения необходимой операции. В применении к 8-разрядным МК процессор с CISC-архитектурой может иметь однобайтовый, двухбайтовый и трехбайтовый (редко четырехбайтовый) формат команд. Время выполнения команды может составлять от 1 до 12 циклов. К МК с CISC-архитектурой относятся МК фирмы Intel с ядром MCS-51, которые поддерживаются в настоящее время целым рядом производителей, МК семейств НС05, НС08 и НС11 фирмы Motorola и ряд других.

В процессорах с RISC-архитектурой набор исполняемых команд сокращен до минимума. Для реализации более сложных операций приходится комбинировать команды. При этом все команды имеют формат фиксированной длины (например, 12, 14 или 16 бит), выборка команды из памяти и ее исполнение осуществляется за один цикл (такт) синхронизации. Система команд RISC-процессора предполагает возможность равноправного использования всех регистров процессора. Это обеспечивает дополнительную гибкость при выполнении ряда операций. К МК с RISC-процессором относятся МК AVR фирмы Atmel, МК PIC16 и PIC17 фирмы Microchip и другие.

На первый взгляд, МК с RISC-процессором должны иметь более высокую производительность по сравнению с CISC МК при одной и той же тактовой частоте внутренней магистрали. Однако на практике вопрос о производительности более сложен и неоднозначен.

С точки зрения организации процессов выборки и исполнения команды в современных 8-разрядных МК применяется одна из двух уже упоминавшихся архитектур МПС: фон-неймановская (принстонская) или гарвардская.

Основное преимущество архитектуры Фон-Неймана - упрощение устройства МПС, так как реализуется обращение только к одной общей памяти. Кроме того, использование единой области памяти позволяло оперативно перераспределять ресурсы между областями программ и данных, что существенно повышало гибкость МПС с точки зрения разработчика программного обеспечения. Размещение стека в общей памяти облегчало доступ к его содержимому. Неслучайно поэтому фон-неймановская архитектура стала основной архитектурой универсальных компьютеров, включая персональные компьютеры.

Гарвардская архитектура почти не использовалась до конца 70-х годов, пока производители МК не поняли, что она дает определенные преимущества разработчикам автономных систем управления.

Дело в том, что, судя по опыту использования МПС для управления различными объектами, для реализации большинства алгоритмов управления такие преимущества фон-неймановской архитектуры как гибкость и универсальность не имеют большого значения. Анализ реальных программ управления показал, что необходимый объем памяти данных МК, используемый для хранения промежуточных результатов, как правило, на порядок меньше требуемого объема памяти программ. В этих условиях использование единого адресного пространства приводило к увеличению формата команд за счет увеличения числа разрядов для адресации операндов. Применение отдельной небольшой по объему памяти данных способствовало сокращению длины команд и ускорению поиска информации в памяти данных.

Кроме того, гарвардская архитектура обеспечивает потенциально более высокую скорость выполнения программы по сравнению с фон-неймановской за счет возможности реализации параллельных операций. Выборка следующей команды может происходить одновременно с выполнением предыдущей, и нет необходимости останавливать процессор на время выборки команды. Этот метод реализации операций позволяет обеспечивать выполнение различных команд за одинаковое число тактов, что дает возможность более просто определить время выполнения циклов и критичных участков программы.

Большинство производителей современных 8-разрядных МК используют гарвардскую архитектуру. Однако гарвардская архитектура является недостаточно гибкой для реализации некоторых программных процедур. Поэтому сравнение МК, выполненных по разным архитектурам, следует проводить применительно к конкретному приложению.

В настоящее время наиболее яркими представителями из компаний микроконтроллеров SISC и RISC, имеющих соответственно фон-неймановскую и гарвардскую архитектуры являются микроконтроллеры i8051 и AVR - микроконтроллеры фирмы Atmel, которые по ряду характеристик превзошли очень известные PIC - микроконтроллеры.

* 1. Принцип действия детектора шума

Приборы для объективного измерения уровня звука носят название шумомеры.

Шумомеры являются измерительными приборами, которые применяются для преобразования колебаний звукового давления в показания соответствующие уровню звукового давления.

Шумомер содержит ненаправленный микрофон, усилитель, корректирующие фильтры, детектор, интегратор (для интегрирующих шумомеров) и индикатор.

Фактически шумомер представляет собой микрофон, к которому подключен вольтметр, отградуированный в децибелах. Поскольку электрический сигнал на выходе с микрофона пропорционален исходному звуковому сигналу, прирост уровня звукового давления, воздействующего на мембрану микрофона, вызывает соответствующий прирост напряжения электрического тока на входе в вольтметр, что и отображается посредством индикаторного устройства, отградуированного в децибелах.

1. ПРАКТИЧЕСКАЯ ЧАСТЬ
   1. Необходимые платы для создания устройства

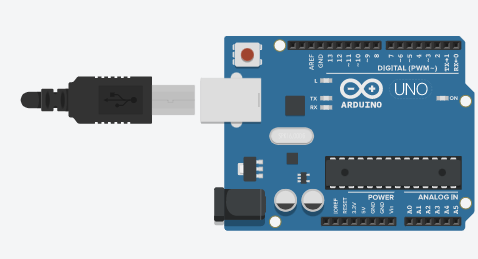
Для создания детектора шума была использована плата Arduino Uno.

Рисунок 1 – Плата Arduino Uno.

Arduino Uno – это полнофункциональное устройство, адаптированное для использования с макетными платами, построенное на базе микроконтроллера (МК) ATmega 328. Основными элементами являются: 14 цифровых портов ввода/вывода, 6 аналоговых входов (от A0 до A5), кварцевый резонатор (генератор тактов) частотой 16 мегагерц (слева от кнопки сброса), разъем USB для подключения к компьютеру, несколько выводов питания, кнопка перезагрузки и сам МК.

Так же понадобятся следующие структурные элементы:

1. Светодиод.

Светодиод сообщает о превышении допустимого значения уровня шума.

Рисунок 2 – Светодиод.

1. Датчик звука.

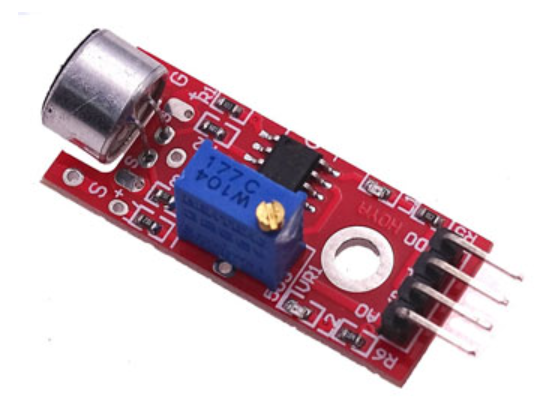
 Датчик выдает аналоговый сигнал, пропорциональный уровню громкости окружающего шума.

Рисунок 3 – Датчик звука.

1. Дисплей.

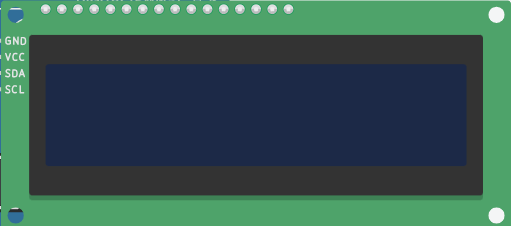
На дисплей выводится значение уровня шума с датчика звука.

Рисунок 4 – Дисплей.

* 1. Подключение элементов

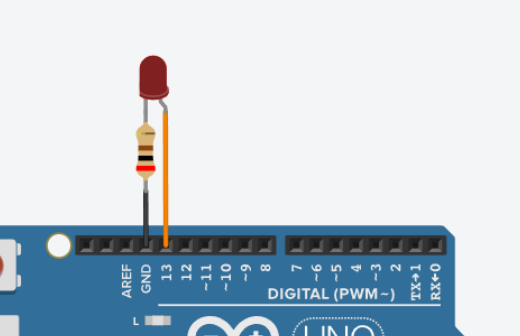
Подключение светодиода. Подсоединяем длинную ножку (анод) к плюсу, т.е. к тому участку схемы, который ведет к цифровому или аналоговому разъему платы. Короткую ножку (катод) через резистор подключаем к минусу – к пину GND.

Рисунок 5 – Подключение светодиода к Arduino Uno.

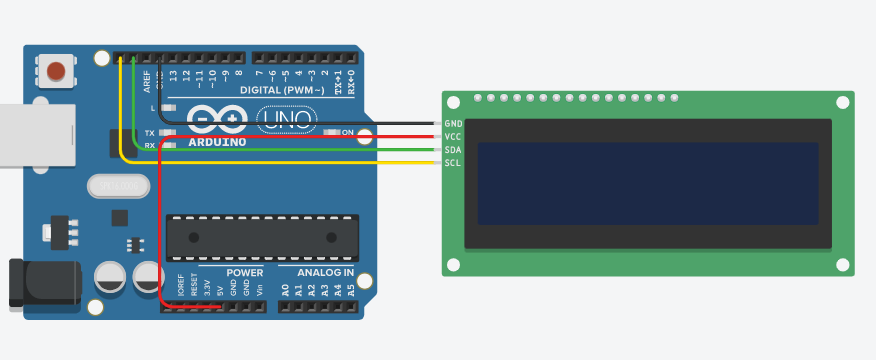
Подключение дисплея. Жидкокристаллический дисплей 1602 с I2C модулем подключается к плате Ардуино всего 4 проводами — 2 провода данных и 2 провода питания. Подключение QAPASS 1602a к Arduino проводится стандартно для шины I2C: вывод SDA подключается к порту A4, вывод SCL – к порту A5. Питание LCD дисплея осуществляется от порта +5V.

Рисунок 6 – Подключение LCD 16x2 к Arduino Uno.

Подключение датчика звука. Питание датчика производится от 5V, выход (OUT, S или AO) подключается к любому аналоговому входу на Arduino Uno, а выход DO к Pin 2, для получения цифрового сигнала на Ардуино с датчика микрофона.

Рисунок 7 – Подключение датчика звука к Arduino Uno.

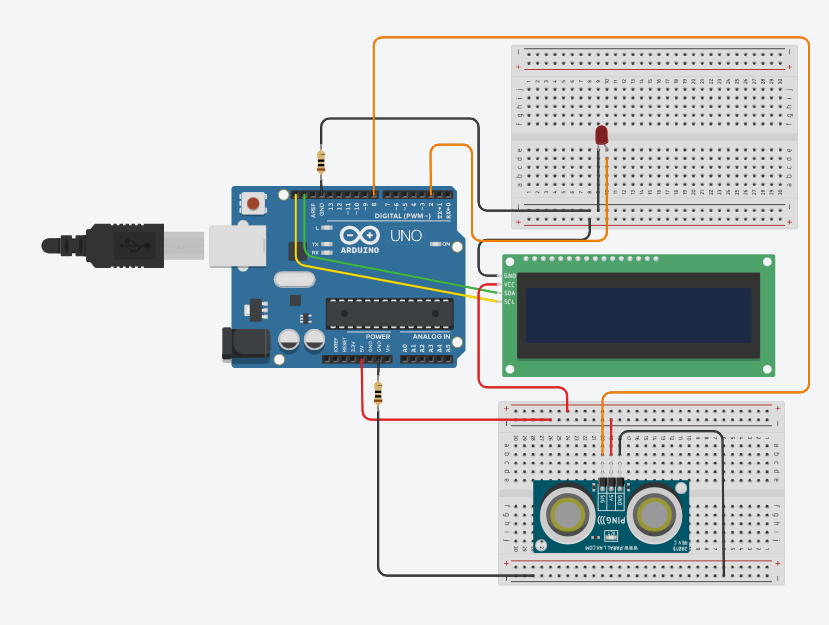


Рисунок 8 – Полная схема подключения элементов.

* 1. Программная составляющая

Для работы с дисплеем необходимо подключить библиотеку Adafruit\_LiquidCrystal.h.

Код программы:

#include <Adafruit\_LiquidCrystal.h> //подключение библиотеки

Adafruit\_LiquidCrystal lcd\_1(0); //присваиваем имя LCD дисплею

void setup()

{

pinMode(7, INPUT); //пин подключения светодиода

lcd\_1.begin(16, 2); //подключение дисплея

}

void loop()

{

if(analogRead(8)> 40) //если значение датчика звука превышает допустимое

digitalWrite(2, HIGH); //то загорается светодиод

lcd\_1.print(analogRead(8)); //вывод на дисплей значение датчика звука

delay(500); //задержка

lcd\_1.setCursor(0, 1); //переход на следующую строку дисплея

}

ЗАКЛЮЧЕНИЕ

Таким образом основываясь на изученной нами теории об микропроцессорах их схемотехнике и их назначении, а также требованиям необходимым для корректной работы детектора шума мы выбрали подходящую микропроцессорную плату Arduino, необходимые для нее элементы для получения и передачи информации. Основываясь на нашем выборе, мы спроектировали детектор шума.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Гоноровский, И.С. Ардуино и программирования: Учебник для вузов. - 4-е изд., перераб. и доп. - М.: Программирование, 2012. - 512 с.

2. И.В. Коршуна. Современные мощные микроконтроллеры: Архитектура, средства проектирования, примеры применения, ресурсы сети Интернет. - М: Аким, 2009, - 272 с.

3. http://habrahabr.ru/ - ресурс IT специалистов

4. http://arduino.ru/ - программирование на ardiuno

5. http://arduino-projects.ru // - сбор проектов arduino