

**МИНЕСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ
УЗБЕКИСТАН
НУКУССКИЙ ФИЛИАЛ ТАШКЕНТСКОГО УНИВЕРСИТЕТА
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
ИМЕНИ МУХАММАДА АЛЬ-ХОРЕЗМИ**



Направление: _____

Курсовая Работа

По предмету _____

На тему _____

Выполнил(а):

Принял(а):

Идриси М.

НУКУС 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	2
1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ	3
1.1. Программируемые отладочные платы семейства Arduino.....	3
1.2. Отладочная плата Arduino Mega 2560	6
1.3. Сравнительная характеристика Arduino Mega2560, Orange Pi 2 и Iskra JS 13	
1.4. Программирование Arduino	18
1.5. Среда программирования Arduino IDE. Язык Arduino.....	18
2. ПРОЕКТИРОВАНИЕ УСТРОЙСТВА.....	25
2.1. Элементная база устройства измерения расстояния на основе программируемой отладочной платы Arduino Mega.....	25
2.2. Разработка устройства измерения расстояния.....	27
2.3. Разработка корпуса для проектируемого устройства.....	28
3. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ	31
3.1. Принцип работы устройства измерения расстояния	31
3.2. Сравнение показаний образцового устройства и разработанного	37
3.3. Расчет основных метрологических характеристик	39
4. Заключение	
5.Список литературы	
6. Приложений	

ВВЕДЕНИЕ

Измерение расстояния необходимо во многих случаях: как в повседневной жизни, так и в производственной отрасли. Данные разработки позволяют создавать бюджетное измерительное устройство, которое не будет уступать своим аналогам в данном сегменте приборов. Разработки такого рода имеют смысл, поскольку созданное устройство не только можно применять в широком диапазоне задач, но и в процессе его создания и модификаций необходимо использовать полученные междисциплинарные навыки. Созданный прибор способен проводить измерения с радиусом в 180 градусов, и выводить результат измерения на дисплей. Для устройства предусматривается возможность управления кнопкой. Также предусматривается возможность перепрограммирования с помощью персонального компьютера, легкость новых модификаций, подключение различных источников питания, а также простота замены элементов устройства при выходе их из строя.

Основой создания измерительного устройства является правильный подбор элементной базы, после чего будет создана схема устройства. Далее предстоит сборка макета устройства по схеме, разработанной ранее и проверка его работоспособности. Одним из главных шагов в процессе разработки является программирование устройства для его корректной работы. После проверки работоспособности, макет будет разобран и изготовлен итоговый вариант измерительного устройства с заменой временных соединений на пайку. Под размеры готового устройства необходимо разработать корпус, который будет защищать от внешних воздействий.

Причина, по которой был выбран ультразвук как способ измерения расстояния – его широкий спектр практического применения. Ультразвук применяется во многих сферах жизни: в медицине, на производстве, в науке, а также в повседневной жизни. В наше время ультразвук применяется в

различных методах и технологиях, например, по скорости распространения звука в среде можно определить ее физические характеристики.

Таким образом, целью выпускной квалификационной работы является моделирование и создание устройства измерения расстояния.

1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Программируемые отладочные платы семейства Arduino

Arduino представляет собой небольшую плату, с размещенными на ней компонентами, главным из которых является микроконтроллер семейства AVR ATmega различных конфигураций. Он является основной вычислительной системой этой платформы, поскольку именно для него и создается программное обеспечение, с помощью которого микроконтроллер взаимодействует с внешним миром посредством специальных портов ввода/вывода данных. Данный продукт, состоит из экономически выгодных и доступных компонентов и главная его задача состоит в том, чтобы гарантировать работу устройства по принципу «plug-and-play»(подключи и используй), — то есть, чтобы пользователь, достав плату из коробки и подключив к компьютеру, мог немедленно приступить к работе. Управление микроконтроллером пользователь осуществляет непосредственно через программный код. Таким образом, любая плата Arduino прежде всего, микроконтроллер, выводы которого удобно разведены по краям платы и подписаны. В таблице 1 приведены основные характеристики ряда плат Arduino, в таблице 2 – характеристики каналов связи платы с компьютером и внешними компонентами. Данные, указанные в таблицах 1 и 2 взяты с официального сайта разработчиков семейства плат и из технических документаций микроконтроллеров[1].

Понятия, используемые в таблицах 1 и 2:

1. Electrically Erasable Programmable Read-Only Memory (EEPROM) – электрически стираемая энергозависимая память с возможностью перепрограммирования
2. Flash-память – энергозависимая память, которая допускает многократную перезапись всего содержимого
3. Static Random Access Memory (SRAM) – статическая энергозависимая память с произвольным доступом
4. Pulse-Width Modulation (PWM) – широтно-импульсная модуляция (ШИМ) – управляет средним значением напряжения на нагрузке через изменения скважности импульсов
5. Universal Asynchronous Receiver/Transmitter (UART) – универсальный асинхронный приемопередатчик

Таблица 1 – технические характеристики программируемых плат Arduino

Наименование	Модель процессора	Входное/рабочее напряжение	Частота процессора	Количество цифровых выводов(ИО/PWM)	Количество аналоговых выводов (In/Out)
Mega 2560	ATmega2560	5В/7-12В	16 МГц	54/15	16/0
Uno	ATmega328	5В/7-12В	16 МГц	14/06	6/0
Nano	ATmega328	5В/7-9В	16 МГц	14/06	8/0
	ATmega168				
Ethernet	ATmega328	5В/7-12В	16 МГц	14/04	6/0
Due	AT91SAM3	3.3В/7-12В	84 МГц	54/12	12/02

	X8E				
Robot	ATmega32u 4	5B/5B	16 МГц	5/6	8/0
M0	ATSAMD21 G18	3.3B/5-15B	48 МГц	20/12	6/0
Pro	ATmega328	3.3B/5-12B	8 МГц	14/6	6/0
Lilypad	ATmega32u 4	3.3B/3.8-5B	8 МГц	9/4	4/0

Продолжение таблицы 1

Gemma	ATtiny85	3.3B/7-12B	8 МГц	03/02	1/0
Tian	SAMD21G1 8	3.3B/3.3-5B	48 МГц	12/12	6/0

Таблица 2 – Характеристики связи и памяти программируемых плат Arduino

Наименование	SRAM (Кбайт)	EEPROM (Кбайт)	Flash (Кбайт)	UART	USB
Mega 2560	8	4	256	4	Regular
Uno	2	1	32	1	Regular
Nano	1	0.512	16	1	Mini-B
	2	1	32		
Ethernet	2	1	32	1	Regular
Due	96	-	32	1	Regular
Robot	2.5	1	32	-	-
M0	32	-	256	1	Micro
Pro	2	1	32	1	-

Lilypad	2.5	1	32	1	-
Gemma	0.5	0.5	8	0	Micro
Tian	32	-	256	1	Regular

На основе указанных в таблицах 1 и 2 данных выбирается плата Arduino Mega 2560. Данная программируемая плата является расширенной платой семейства Arduino, имеет увеличенные размеры по сравнению с другими платами (10,16 × 5,3 см против 6,9 × 5,3 см базовой модели). Эта отладочная плата имеет стандартное выходное напряжение – 5 В и удобный рабочий диапазон от 7 до 12 В). Arduino Mega2560 имеет 54 цифровых и 16 аналоговых выводов, что позволяет проектировать устройства с большим количеством модулей и плат расширений. Подключение данной платы к компьютеру происходит через USB-интерфейс. Объем flash-памяти равен 256 Кбайт, что выше стандартного объема среди остального ряда плат. Всех этих характеристик будет достаточно, чтобы реализовать идею проекта.

Для решения поставленной задачи будет использована ранее рассмотренная отладочная плата и разработка будет актуальна только для Arduino Mega 2560.

1.2. Отладочная плата Arduino Mega 2560

Arduino Mega2560, представленная на рисунках 1 и 2 – программируемая отладочная плата на основе AVR микроконтроллера ATmega2560, принципиальную схему которой можно найти в приложении А. В состав платы входят: 6 аналоговых входов, 54 цифровых выводов (15 из которых могут использоваться в качестве ШИМ-выходов), кварцевый резонатор на 16 МГц, разъемы USB, питания, внутрисхемного программирования и кнопка сброса. Для вывода 3,3 В максимальный входной ток равен 50 мА, а для вывода 5 В – 40 мА.



Рисунок 1 – Отладочная плата Arduino Mega 2560 (лицевая сторона)

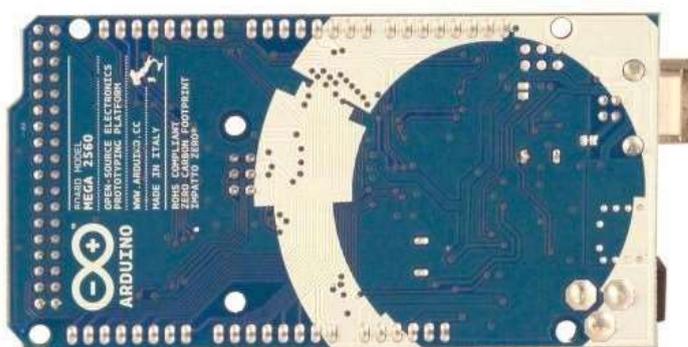


Рисунок 2 – Отладочная плата Arduino Mega 2560 (обратная сторона)

Для запуска отладочной платы ее необходимо запитать. Сделать это можно двумя способами: либо от USB, либо от внешнего источника питания. В качестве внешнего источника питания возможно использовать сетевой AC/DC-адаптер или аккумулятор/батарею. При подключении источника питания непосредственно к портам Arduino необходимо соединить провода к выводам Gnd и Vin разъема POWER.

Заявленное напряжение внешнего источника питания может быть в пределах от 6 до 20 В, однако для поддержания стабильной работы устройства на выводе 5 В не желательно уменьшать напряжение питания ниже 7 В. Так же не рекомендуется повышение напряжения больше 12 В во

избежание перегрева стабилизатора напряжения и выхода платы из строя. Из всего этого следует, что рекомендованный диапазон напряжения для источника питания составляет от 7 до 12 В.

Выводы питания, расположенные на плате:

1. IOREF. Вывод, с которого поступает информация о рабочем напряжении микроконтроллера для плат расширения. В зависимости от напряжения, считанного с вывода IOREF, плата расширения может задействовать преобразователи уровней для работы как с 3,3 В, так и с 5,5 В устройствами
2. GND – земля
3. 3.3V – 3,3 В, поступающие от стабилизатора напряжения на плате
4. 5V – на вывод поступает напряжение 5 В от стабилизатора платы, не зависимо от типа питания устройства. Не рекомендуется запитывать устройство через выводы 3.3V и 5V, так как в таком случае стабилизатор не используется, что чревато выходом платы из строя.
5. VIN – напряжение, поступающее на плату от внешнего источника питания. Через этот вывод помимо подачи внешнего питания можно потреблять ток, когда устройство запитано от внешнего адаптера.

В центре платформы Arduino Mega 2560 располагается 8-битный микроконтроллер семейства AVR – ATmega2560. Он предоставляет 256 КБ флеш-памяти для хранения прошивки, 8 КБ оперативной памяти SRAM и 4 КБ энергонезависимой памяти EEPROM для хранения данных.

Все 54 цифровых вывода можно настраивать на вход или выход используя функции `pinMode()`, `digitalWrite()`, и `digitalRead()`. Выводы работают при напряжении 5 В. Каждый вывод имеет нагрузочный резистор (стандартно отключен) 20-50 кОм и может пропускать токи до 40 мА.

Большинство выводов Arduino способны выполнять дополнительные функции:

1. Последовательный интерфейс: выводы 0, 19, 17 используются для

получения данных по последовательному интерфейсу (RX), выводы 1, 18, 16 применяются для передачи данных по последовательному интерфейсу (TX). Выводы подключены к соответствующим выводам микросхемы последовательной шины ATmega8U2, являющейся преобразователем USB-UART.

2. Внешние прерывания: выводы 2, 3, 18, 19, 20, 21. Данные выводы могут использоваться на вызов прерывания либо на младшем значении, либо на переднем или заднем фронте, или при изменении значения. Более подробная информация находится в описании функции `attachInterrupt()`.
3. ШИМ(PWM): выводы с 2 до 13 и 44-46. Любой из этих выводов обеспечивает широтно-импульсную модуляцию с разрешением 8 бит при использовании функции `analogWrite()`.
4. SPI-интерфейс: выводы 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Посредством данных выводов осуществляется связь SPI.
5. Светодиод: вывод 13. Встроенный светодиод, подключенный к выводу 13. Если на выводе сигнала значение LOW, светодиод выключен, при отправке HIGH – светодиод горит.
6. I2C-интерфейс: Выводы 20(SDA) и 21(SCL). Через эти выводы осуществляется связь I2C. Для осуществления связи необходимо подключить дополнительную библиотеку `Wire`.

Кроме основных вышеперечисленных, на плате существует дополнительная пара выводов:

1. Reset. Образует низкий уровень сигнала на выводе, что приводит к перезагрузке микроконтроллера. Чаще всего этот вывод служит для подключения кнопки перезагрузки на плате расширения, если доступ к кнопке на плате закрыт.
2. AREF. Опорное напряжение для аналоговых выводов, для применения функции `analogReference()`.

Платформа Mega 2560 обладает 16 аналоговыми входами (A0-A15), каждый разрешением 10 бит (т.е. может принимать 1024 различных

значения). По умолчанию рабочий диапазон измерения составляет до 5 В относительно земли, однако существует возможность изменить верхний предел через вывод Analog Reference (AREF) с использованием функции `analogReference()`. Наглядное представление всех вводов и выводов на рисунке 3.

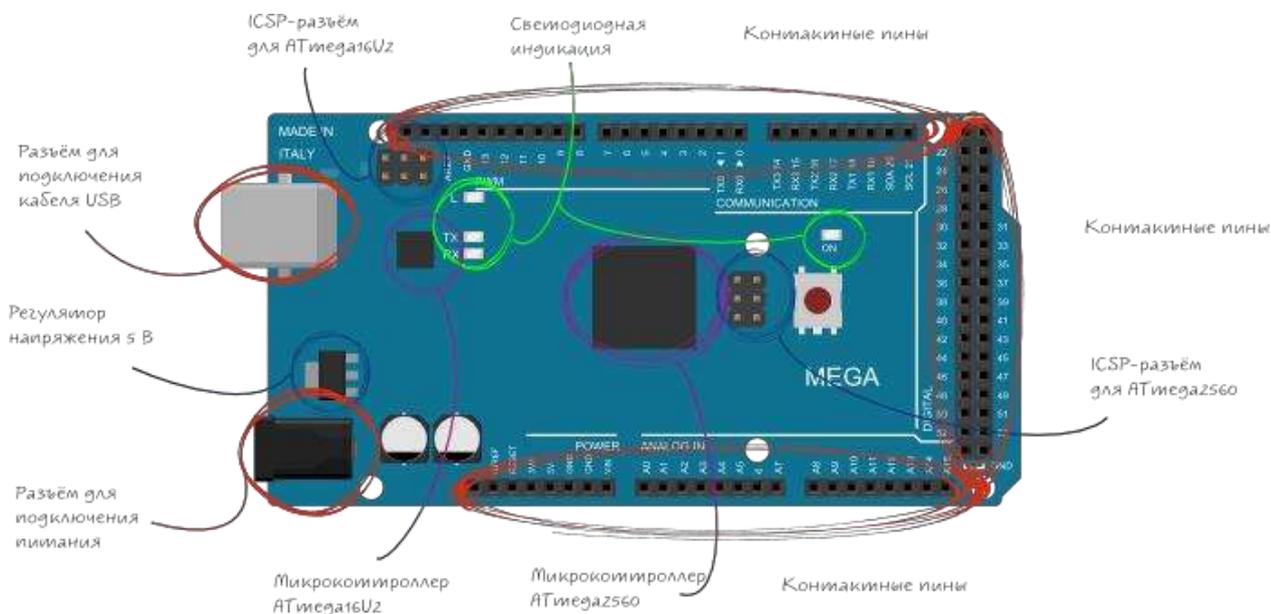


Рисунок 3 – Наглядное представление расположения портов на Arduino Mega 2560

Для осуществления связи с компьютером Arduino Mega 2560 обладает несколькими вариантами подключения: микроконтроллерами или другими устройствами Arduino. ATmega 2560 поддерживает 4 порта последовательной передачи данных UART, которые предназначены для последовательной связи посредством цифровых выводов. Микроконтроллер ATmega16U2 осуществляет связь приемопередатчика с USB-портом компьютера, предоставляя виртуальный COM (communications) – порт программ на компьютеры. Машинам под управлением ОС Windows для корректной работы с виртуальным COM портом необходим файл расширения `.inf`, системы на базе OSX и Линукс автоматически распознают COM порт. Специальная утилита мониторинга последовательной шины (Serial Monitor), входящая в пакет программного обеспечения Arduino позволяет посылать и получать текстовые данные при подключении к платформе. Во время USB

соединения при передаче данных через микросхему ATmega8U2 будут гореть светодиоды RX и TX. При работе последовательной передачи через выводы 0 и 1 без использования USB-преобразователя указанные ранее светодиоды не задействуются.

С помощью библиотеки SoftwareSerial существует возможность создать последовательную передачу через любой из цифровых выводов Mega 2560.

Помимо прочего микроконтроллер ATmega2560 способен работать с интерфейсами I2C (TWI) и Serial Peripheral Interface (SPI). Для удобства работы с этими интерфейсами в программное обеспечение Arduino добавлена библиотека Wire, позволяющая упростить работу с шиной I2C. Для связи с интерфейсом SPI используется библиотека SPI.

Программирование Arduino Mega 2560 происходит с помощью программного обеспечения Arduino Integrated Development Environment (IDE), которое является интегрированной средой разработки. Написание кода программы происходит на языках Processing/Wiring предназначенных для плат Arduino и являющихся разновидностью языка C/C++.

ATmega2560 в Arduino Mega 2560 обладает прошитым загрузчиком, с помощью которого в микроконтроллер загружаются новые программы без использования внешнего программатора.

Тем не менее, можно прошить микроконтроллер через разъем внутрисхемного программирования ICSP (In-Circuit Serial Programming), игнорируя загрузчик.

Код исходной прошивки для микроконтроллера ATmega8U2 доступен для свободного скачивания, что позволяет прошить микроконтроллер в любой момент. ATmega8U2 обладает собственным DFU-загрузчиком (Device Firmware Update – обновление программного обеспечения устройства), позволяющим обновлять прошивку микроконтроллера. Он может быть активирован путем замыкания джампера на обратной стороне платы (рядом с картой Италии) и перезагрузкой контроллера.

После того, как микроконтроллер переведен в режим DFU, для

обновления прошивки возможно использовать программное обеспечение Atmel's FLIP (для Windows) или DFU-программатор (для Mac OS и Linux). Также существует возможность альтернативной записи через вход ISP (разъем внутрисхемного программирования), но в таком случае DFU-загрузчик будет затерт.

Для того чтобы каждый раз не требовалось нажимать кнопку сброса перед загрузкой программы, одна из линий ATmega8U2, управляющая потоком данных (DTR), подключена к выводу RESET через конденсатор 100 нФ. Когда линия активизируется, т.е. подается сигнал низкого уровня, вывод RESET также переходит в низкий уровень, и микроконтроллер перезагружается. Данная способность позволяет прошивать микроконтроллер одним нажатием кнопки Upload в самой среде программирования Arduino. Такая архитектура минимизирует таймаут загрузчика, поскольку процесс скоординирован с подачей сигнала низкого уровня на линию DTR[2].

Система содержит следующий недостаток: при подключении Arduino Mega к компьютеру с операционной системой, отличной от Windows, ее микроконтроллер будет сбрасываться при каждом соединении платы и программного обеспечения. Около полсекунды необходимо для Arduino Mega 2560 на активизирование загрузчика. Хотя загрузчик и запрограммирован игнорировать все данные, не касающиеся процесса прошивки новой программы, он способен перехватить несколько первых байт информации из посылки, отправляемой плате сразу после подключения. Чтобы исключить подобные ситуации, необходимо убедиться, что программное обеспечение, с которым взаимодействует плата, начнет отправлять данные не ранее, чем через секунду после установки соединения.

Для отключения линии автоматической перезагрузки необходимо осуществить разрыв соответствующей линии. Делается это через дорожку, отмеченную как «RESET-EN», разомкнув которую будет осуществлён автоматический сброс микроконтроллера. Также для этой цели можно

использовать резистор номиналом 110 Ом, который необходимо подключить между выводом RESET и 5 В. Для возобновления работы функции автоматического сброса будет достаточно спаять между собой выводы, расположенные по краям дорожки.

Для защиты USB-порта компьютера от сверхтоков и токов короткого замыкания Arduino Mega обладает перезагружаемой плавкой вставкой. Несмотря на то, что большинство компьютеров обладают собственной подобной защитой, данный предохранитель обеспечивает дополнительный барьер. При через USB-порт прохождении тока более 500 мА предохранитель автоматически прерывает обмен данных до перезагрузки или короткого замыкания.

В соответствии с техническим руководством для микроконтроллера ATmega2560 предусмотрено опорное напряжение $1,1 \pm 0,1$ В.

Максимальный размер платы составляет $10,16 \times 5,3$ см, а гнезда для USB и внешнего питания выступают за обозначенные границы. Также на плате предусмотрены 6 крепежных отверстий под шурупы или винты. Расстояние между контактами составляет 2,54 мм, но в случае 7-го и 8-го цифрового входа – расстояние: 4 мм.

1.3. Сравнительная характеристика Arduino Mega2560, Orange Pi 2 и Iskra JS

Помимо Arduino существуют также и другие платы, поэтому после выбора Arduino Mega 2560 в данном подразделе рассмотрены платы других производителей с аналогичными характеристиками. Сравнение проводится для:

1. Расширенной версии отладочной платы семейства Arduino: Arduino Mega2560.
2. Платы Orange Pi Plus 2 – одноплатный компьютер, представленный на рисунке 4
3. Платы Strela, представленной на рисунке 5

Причины, по которым были выбраны эти платы – максимально близкие возможности для решения однотипных задач, например, создание умного дома или проектов для автомобилестроения.



Рисунок 4 – Плата Orange Pi Plus 2

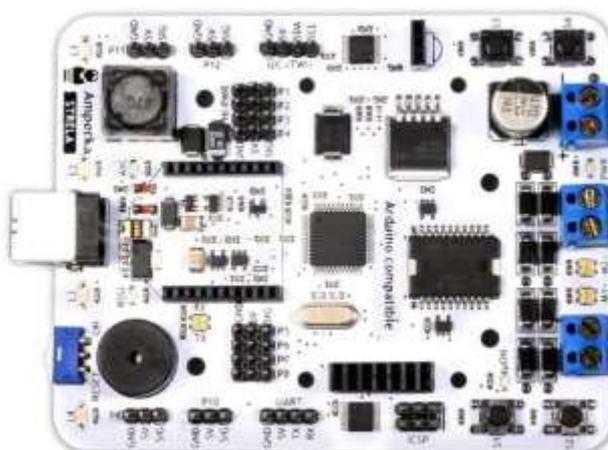


Рисунок 5 – Плата Strela

В таблице 3 представлены краткие характеристики для всех выбранных плат[3].

Таблица 3 – Технические характеристики Arduino Mega 2560, Orange Pi Plus 2, Strela

Платформа	Arduino Mega	Orange Pi Plus 2	Strela
Хар-ка	2560		

Габаритные размеры	10,16 × 5,3 см	9,3 × 6 см	10,5 × 7,9 см
-----------------------	----------------	------------	---------------

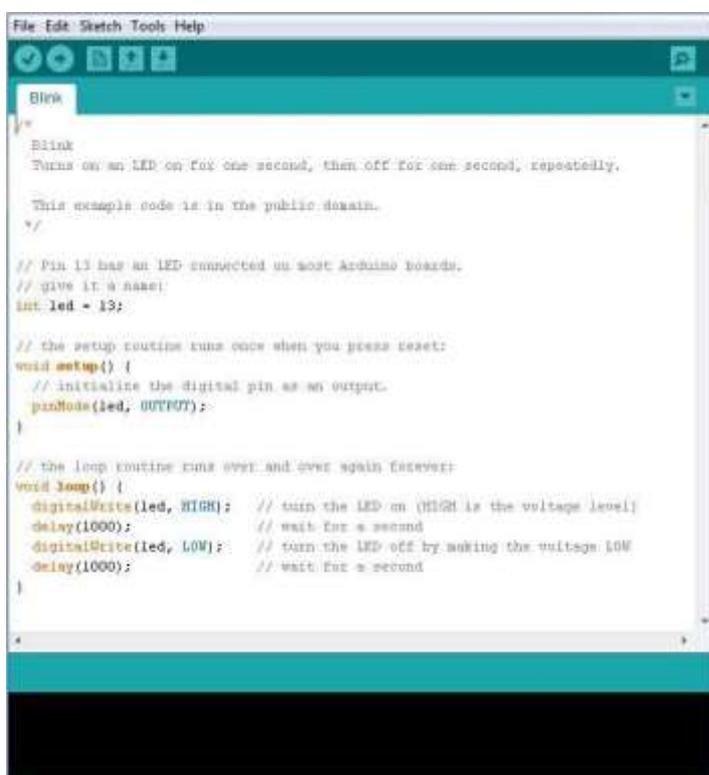
Продолжение таблицы 3

Ориентировочная цена	38\$	72\$	49\$
Микроконтроллер	ATmega2560	ARM Cortex-A7	ATmega32u4
Тактовая частота	16 МГц	1600 МГц	16 МГц
Flash-память	256 Кбайт	MicroSD карта	32 кБ
EEPROM	4 Кбайт	-	1 кБ
ОЗУ	8 Кбайт	2 Гб	2,5 кБ
Минимальное энергопотребление	42 мА (0,3 Вт)	700 мА (3,5 Вт)	42 мА (0,3 Вт)
Напряжение питания	7-12 В	5В	5 В
Цифровые входы/выводы	54	40	16
Аналоговые входы	16	-	-
Каналы ШИМ	14	-	4
Инструменты разработки	Arduino IDE, Visual Studio	Linux, IDLE, Scratch,	Scratch, Arduino IDE, Web IDE
Порт Ethernet	-	10/100M Ethernet RJ45	-
Интерфейс USB Master	-	4 USB 2.0	-

Среди рассмотренных вариантов Arduino является самым бюджетным вариантом, даже с учетом того, что модель Mega 2560 является одной из

самых дорогих среди модельного ряда производителя. А стоимость модулей, сенсоров или шилдов колеблется от 1\$ до 12\$. А для начала работы Orange Pi Plus 2 необходима дополнительная покупка SD-карты, что увеличивает стоимость на 10-15\$.

Для плат Arduino и Strela программный код пишется в среде Arduino IDE, которая показана на рисунке 6. Данная среда основывается на компиляторе языка C/C++. Однако, программирование под плату Strela осложняется обязательным наличием дополнительной библиотеки, которая является любительской и нередко ее работа нестабильна.

The image shows a screenshot of the Arduino IDE software. The window title is "File Edit Sketch Tools Help". The main text area contains the following code:

```
Blink
//
// Blink
// Turns on an LED on for one second, then off for one second, repeatedly.
//
// This example code is in the public domain.
//
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

Рисунок 6 – Графический интерфейс Arduino IDE

Главным отличием Orange Pi Plus 2 является работа под операционной системой Linux, под управлением которой работает данная плата. Благодаря этому программному обеспечению платы такого вида можно считать полноценными компьютерами, способными к многозадачности и которые могут программироваться на нескольких языках. Orange Pi Plus 2 не стоит выбирать новичкам, поскольку программирование в Unix-системах очень

трудоёмкий и сложный процесс. Помимо этого, Arduino обладает огромным сообществом, включающим в себя множество пользователей, большое количество учебных материалов, готовых решений и проектов, которые могут стать подспорьем в разработках.

Бывают проекты, в которых необходимо подключение к сети интернет. Здесь более практично использование Orange Pi Plus 2, поскольку это устройство имеет в наличии порты Ethernet и USB. Через USB-порт возможно подключение модулей беспроводной передачи данных для реализации подключения к сети интернет.

На платформе Arduino также есть возможность для реализации обмена данными по Ethernet с помощью плат расширения (Shield), но качество сетевой функциональности будет заметно ниже и ограничена. Тоже касается и Wi-Fi модуля для беспроводной передачи данных. Таким образом, если в проекте не возникает потребности в высокой сетевой необходимости, то отладочные платы Arduino будут оптимальным выбором.

Платформа Strela обладает характеристиками, схожими с Arduino, однако, эта плата является частной разработкой, а для программирования используется JavaScript. Сам язык программирования JavaScript не лишен несколько недостатков, наиболее значимые из которых[3]:

- Язык компилируется каждый раз при исполнении кода, что ведет к увеличению время выполнения программы
- Отсутствует типизация данных, следовательно, работу строки нельзя проверить, пока выполнение кода не дойдет до нее.
- Классы и их наследование значительно отличаются от привычной реализации в C#/C++.

Arduino – это гибкая и функциональная платформа, подходящая для разработки встраиваемых приложений с огромными возможностями для взаимодействия с внешним миром. Она отлично подойдет для работы с микроконтроллерами и может служить основой для множества идей.

1.4. Программирование Arduino.

Arduino Mega 2560 обладает микроконтроллером ATmega2560, который подлежит программированию для выполнения поставленных задач. Одним из самых популярных языком программирования является непосредственно язык Arduino, обладающий своей интегрированной средой разработки Arduino IDE и созданный производителем плат семейства Arduino. С помощью представленного языка можно разрабатывать программы для управления сервоприводами, опроса датчиков и создание приложений в среде Arduino IDE.

1.5. Среда программирования Arduino IDE. Язык Arduino.

Среда разработки Arduino, представленная на рисунке 7, состоит из следующих элементов :

1. Verify (Проверить): Собирает (компилирует) проект и выполняет проверку на отсутствие ошибок в коде программы.
2. Upload (Загрузить): Загружает программу в микроконтроллер на плате Arduino. Перед этим необходимо убедиться в том, что плата подключена корректно
3. New (Новый): Создает новую программу (так называемый скетч).
4. Open (Открыть): Открывает меню со списком проектов.
5. Save (Сохранить): Сохраняет активный, текущий проект.
6. Serial Monitor (Сериал Монитор): Отображает работу COM порта с текущим проектом и выводит результат работы программы в эту область.
7. Sketch Name (Имя скетча): Отображает имя текущего проекта.
8. Code Area (Область кода): Область кода программы.
9. Message Area (Область сообщений): В эту область выводятся различные сообщения, в том числе и об ошибках



Рисунок 7 – Элементы графического интерфейса Arduino IDE

Интегрированная среда разработки Arduino IDE создана производителем плат семейства Arduino, что позволяет моментально выбрать плату, под которую пишется программа. Через компилятор существует возможность выбрать порт для подключения платы, скомпилировать и загрузить скетч (программу) в микроконтроллер, установленный на плате. Одной из особенностей такого программирования является большое количество различных библиотек, находящихся в свободном доступе и позволяющих работать с различными модулями, датчиками или платами расширения. Данный подход значительно упрощает процесс подключения сторонней аппаратуры и написание скетча для его ее стабильной работы. Среда разработки и драйвера для плат созданы одной компанией, что позволяет использовать данную интегрированную среду не только на различных версиях операционной системы Windows, но и на Linux и Mac OS,

обладая полной совместимостью с этими операционными системами. Для того, чтобы связь между стационарным компьютером (ноутбуком) и отладочной платой была установлена, необходимо подключить их кабелем USB, идущем в комплекте с платой, никаких дополнительных средств не требуется. Микроконтроллер Arduino обладает прошитым загрузчиком, позволяющим загружать в микроконтроллер новые программы без использования программатора со стороны.

Основой для программирования устройств Arduino является язык C/C++, что делает его популярным, и максимально удобным для программирования устройств, основанных на микроконтроллерах.

Язык Arduino включает:

1. Операторы

- 1.1. Операторы управления (операторы условия «if», «if...else», циклов «while», «for», «do...while» и другие)
- 1.2. Синтаксис (знаки препинания «;» «{ }», объявление комментариев «//» «/* */», операторы включения библиотек «#include»)
- 1.3. Операторы сравнения («!=» - не равно, «==» - равно» и другие)
- 1.4. Арифметические операторы
- 1.5. Логические операторы («||» - ИЛИ, «&» - И, «!» - отрицание)
- 1.6. Битовые операторы («~» - побитовое НЕ, «&» - побитовое И, «>>» - побитовый сдвиг вправо и другие)
- 1.7. Составные операторы («--» - декремент, «++» - инкремент, «+=» - составное сложение и другие)
- 1.8. Указатели доступа («&» - ссылки, «*» - указатели)

2. Данные

- 2.1 Константы (Зарезервированные – «HIGH», «INPUT», «true»,

целочисленные, с плавающей точкой и другие)

2.2 Типы данных (целочисленные «int», «short», «long», вещественные «float», логические «boolean», символьные «char», массивы и другие)

2.3 Преобразование типов данных («float()», «char()», «byte()» и другие)

2.4 Область видимости переменных и квалификаторы («const», «volatile», «static»)

2.5 Вспомогательная функция, которая возвращает количество байт, занимаемых массивом или типом переменной («sizeof()»)

3. Функции

3.1 Аналоговый ввод/вывод («analogRead()», «analogWrite()»)

3.2 Цифровой ввод/вывод («digitalRead()», «digitalWrite()», «pinMode()»)

3.3 Расширенный ввод/вывод («tone()», «pulseIn()»)

3.4 Время (возвращение прошедшего времени со старта программы «millis()», «micros()», приостановка выполнения программы «delayMicroseconds()»)

3.5 Тригонометрические функции («cos()», «sin()», «tan()»)

3.6 Математические функции («max()», «abs()», «sq()» и другие)

3.7 Генератор случайных чисел («random()»)

3.8 Функция передачи данных («Serial»)

3.9 Внешние прерывания («attachInterrupt()», «detachInterrupt()»)

3.10 Прерывания («interrupts()»)

3.11 Биты и байты (извлечение старшего/младшего байта переменной

«highByte()», «lowByte()», очистка/установка значения бита «bitClear()», «bitSet()» и другие)

4. Библиотеки

4.1 Servo – библиотека управления сервоприводами

4.2 EEPROM – чтение и запись энергозависимой памяти микроконтроллера

4.3 SPI – библиотека, реализующая передачу данных через интерфейс SPI

4.4 Stepper – библиотека управления шаговыми двигателями

Все это доказывает, что язык Arduino является полноценным языком программирования, который обладает привычным синтаксисом и большим количеством библиотек для корректной работы стороннего оборудования. Однако, как и в любом другом языке программирования, в языке Arduino каждая из операций может выполняться в течение длительного времени, что не всегда удобно. Для решения этой проблемы в микроконтроллерах ATmega предусмотрена возможность управления портами через регистры. Соответствие выводов микроконтроллера и платы показано на рисунке 8[4].

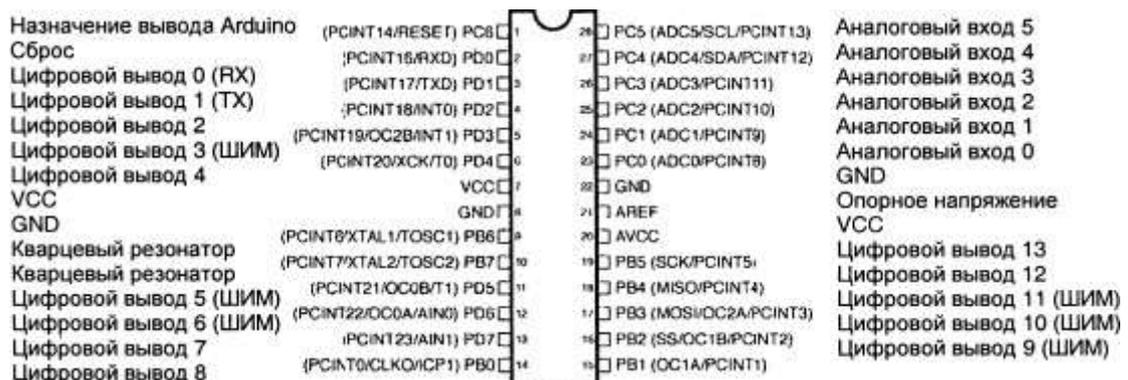


Рисунок 8 – Соответствие выводов в ATmega2560 и платы Arduino.

Описание регистров для микроконтроллера ATmega2560:

- PORTC отображается на аналоговые выводы от 0 до 5

- PINC – адрес входных выводов порта C
- PORTC – регистр данных порта C
- DDRC – регистр направления передачи данных порта C
- PORTD отображается на цифровые выходы от 0 до 7
 - PIND – адрес входных выводов порта D, одновременно считывает все цифровые входы
 - PORTD – регистр данных порта D
 - DDRD – регистр направления передачи данных порта D
- PORTB отображается на цифровые выходы от 8 до 13
 - PINB – адрес входных выводов порта B
 - PORTB – регистр данных порта B
 - DDRB – регистр направления передачи данных порта B

Стоит учесть, что выходы 0 и 1 предназначены для последовательного обмена при программировании и отладке платы Arduino, так что рекомендуется избегать изменения состояния этих регистров, кроме случаев, когда необходимо задействовать функции последовательного ввода/вывода. Для направления передачи данных порта D предназначен регистр DDRD. Биты данного порта управляют тем, настроены ли выходы порта D как выходы, или как входы, пример такого использования показан на рисунке 9:

```

DDRD = B11111110; // назначение выводов 1-7 для выхода, а вывода 0 - для ввода
DDRD = DDRD | B11111100; // назначение выводов со 2 по 7,
                        //оставляя значения на выводах 0 и 1 для отладки

```

Рисунок 9 – пример использования регистра DDRD

Пример использования этого же способа для регистра состояния выходов PORTD продемонстрирован на рисунке 10:

```
PORTD = B10011000; //установка состояния HIGH на цифровых выводах 7,4,3
```

Рисунок 10 – Пример использования регистра PORTD

Таким образом, получим 5В на этих выводах только в случае, когда они были назначены выходными через регистр DDRD или с помощью функции pinMode().

Плюсы такого программирования отладочных плат Arduino:

- Данный подход позволяет уменьшить объем кода и соответственно, освободить некое количество памяти. Потребуется меньшее количество байт для одновременного использования ряда аппаратных выводов.
- При необходимости использования нескольких выводов точно в одно время, т.к. использование классических digitalWrite() приведет к тому, что требуемые выводы могут быть установлены с разностью в несколько микросекунд
- Скорость переключения выводов возрастает в несколько раз. digitalWrite() и digitalRead() содержат большое количество строк кода и будут скомпилированы в большое число машинных команд. Прямой доступ к порту делает эту процедуру за меньшее число тактов.

Минусы:

- Код гораздо более сложен, отладка и поддержка может вызвать трудности у других пользователей . Если быстроедействие не является критическим фактором проекта – данным способом обращения можно пренебречь
- Код не так переносим с одной платы на другую, так как управляющие

регистры и регистры порта могут отличаться у каждого типа микроконтроллера. Использование `digitalWrite()` и `digitalRead()` гарантирует работу на всех микроконтроллерах семейства Atmel

2. ПРОЕКТИРОВАНИЕ УСТРОЙСТВА

2.1. Элементная база устройства измерения расстояния на основе программируемой отладочной платы Arduino Mega.

Следующим шагом после изучения принципов работы отладочной платы Arduino и исследования ее свойств является выбор элементной базы для реализации разрабатываемого устройства. Ниже приведено описание всех модулей, которые были использованы при разработке измерительного устройства:

1. Тактовая кнопка, представленная на рисунке 11-1, предназначена для управления измерением, а именно для возможности остановить измерения, чтобы снять результат.
2. Ультразвуковой датчик HC-SR04 изображен на рисунке 11-2. Данный датчик способен измерять расстояние до 4 метров, а одним из главных плюсов является независимость показаний от солнечного света или цвета объекта.
3. Для корректного отображения результата измерения используется двухстрочный LCD дисплей, показанный на рисунке 11-3. Дисплей способен отображать по 16 знаков в каждой строке. В случае необходимости, присутствует возможность управлять подсветкой синего цвета через потенциометр. Подключение и управление осуществляется по шине I2C.
4. Для возможности проведения измерений в диапазоне 180 градусов используется сервопривод TowerPro SG90, представленный на рисунке 11-4. Скорость вращения сервопривода можно регулировать с помощью

программного кода, а максимальная скорость вращения составляет 60 градусов за 0.10 секунд

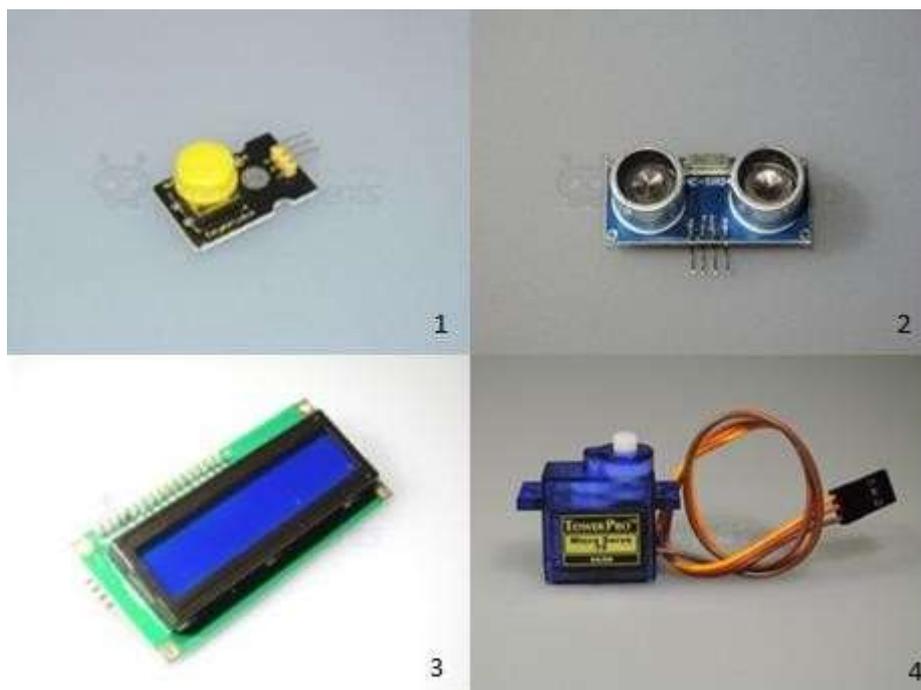


Рисунок 11 – Модули, которые использовались в проектировании устройства

Все модули обладают номинальным напряжением питания от 3.3В до 6В, что удобно, так как эти значения попадают в диапазон напряжений Arduino Mega2560.

Для работы любого электронного устройства должен присутствовать источник питания. Для этой цели подойдет адаптер, подключенный к сети 220В, мобильный источник питания или компьютер, соединенный с платой через USB порт.

Перед тем, как собрать устройство окончательно, запаяв все соединения, необходимо создать макет и проверить его работоспособность.

Для выполнения данной задачи используется макетная плата (breadboard), которая состоит из отверстий-гнезд с шагом 2,54 мм (0,1 дюйма). Размер соответствует шагу выводов современных радиодеталей, что позволяет устанавливать провода диаметром до 0,7 мм и использовать микросхемы в DIP-корпусах (Dual-In-line Package)[5]. Для того чтобы соединить

отладочную плату с макетной платой необходимо использовать провода типа «папа-папа». После того, как макет будет создан, а его работоспособность подтверждена, производится пайка на макетной плате, изготовленной специально под пайку. На рисунке 12 приведена схема подключения всех элементов к плате

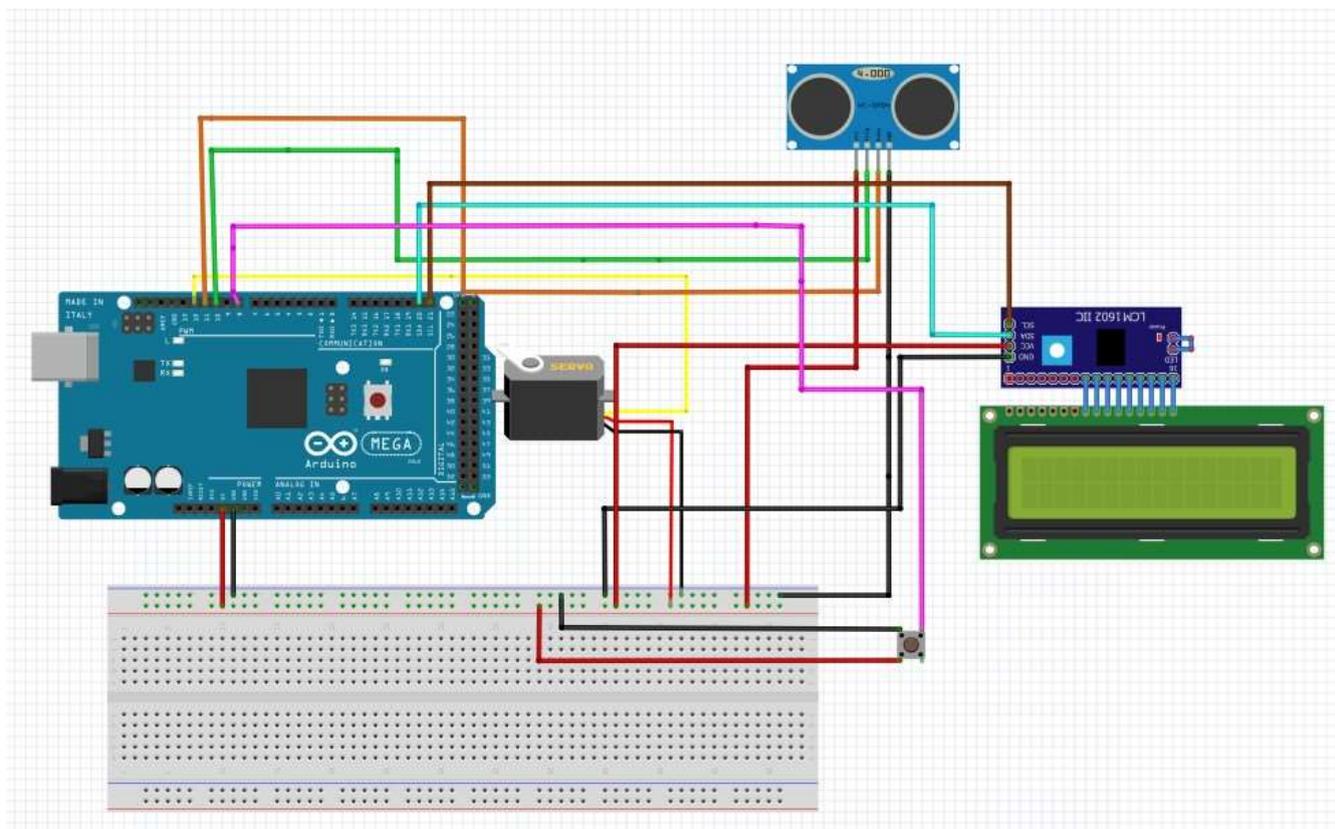


Рисунок 12 – Схема подключения элементов в реализации устройства для измерения расстояния

2.2. Разработка устройства измерения расстояния.

Первый этап в разработке устройства состоит в создании макета устройства измерения расстояния, который показан на рисунке 13. Собранный макет полностью работоспособен, но в нем отсутствует пайка и он может быть модифицирован или разобран без каких-либо сложностей. Создание данного макета дает возможность проверить правильность сборки и стабильной работы разработанного устройства. Помимо этого сборка макета дает возможность оценить работу программы для микроконтроллера

и сравнить показания устройства с дальномером, принятым за образцовый при проверке.

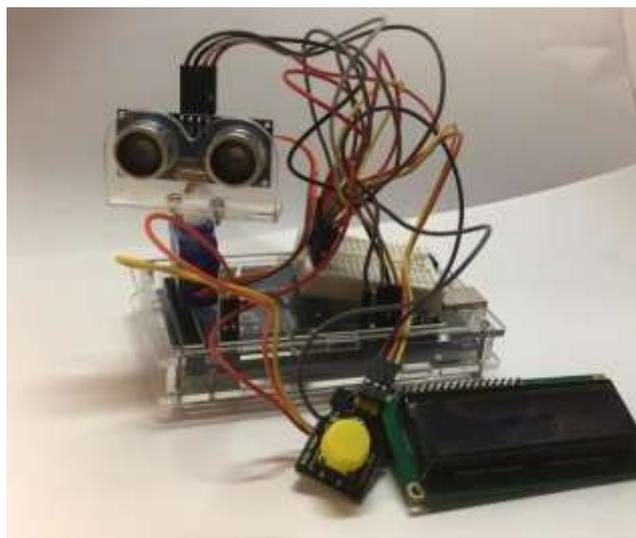


Рисунок 13 – Рабочий макет устройства

Следующим шагом в разработке является пайка всех элементов и соединений на макетной плате, за исключением отладочной платы, для которой сохраняется возможность отсоединения.

2.3. Разработка корпуса для проектируемого устройства.

Третий этап состоит из создания корпуса и установки устройства измерения в корпус. Модель корпуса, показанная на рисунке 14 разработана в системе трехмерного моделирования КОМПАС-3D. Полноценные чертежи корпуса можно найти в приложении Б.

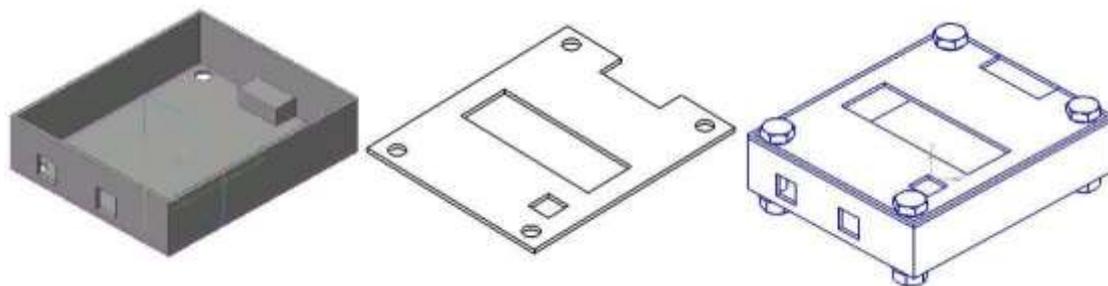


Рисунок 14 – создание корпуса для устройства измерения

Данная система способна генерировать ассоциативные виды трехмерных моделей (включая сечения, разрезы, местные виды и т.д.), которые ассоциированы с моделью, то есть, изменения в модели непосредственно приводят к изменению изображений на чертежах.

КОМПАС-3D широко используется в таких отраслях как машиностроение и приборостроение, так как позволяет проектировать изделия любой сложности и соответствует методикам проектирования[6].

Далее, файл разработанного корпуса направляется на станок с числовым программным управлением, где и до конца изготавливается. Корпус выполнен из авиационной фанеры толщиной 4мм и покрыт несколькими слоями водоотталкивающего прозрачного акрилового лака. Авиационная фанера состоит из нескольких тонких слоев высококачественного березового шпона и обладает хорошей прочностью, легкостью, гибкостью и малой толщиной. Акриловый лак, который наносится на фанеру, позволяет дополнительно защитить корпус от воздействия влаги, температур, ультрафиолета и других внешних факторов. Размеры корпуса: длина – 13 см, ширина – 7 см, высота – 4.5 см.

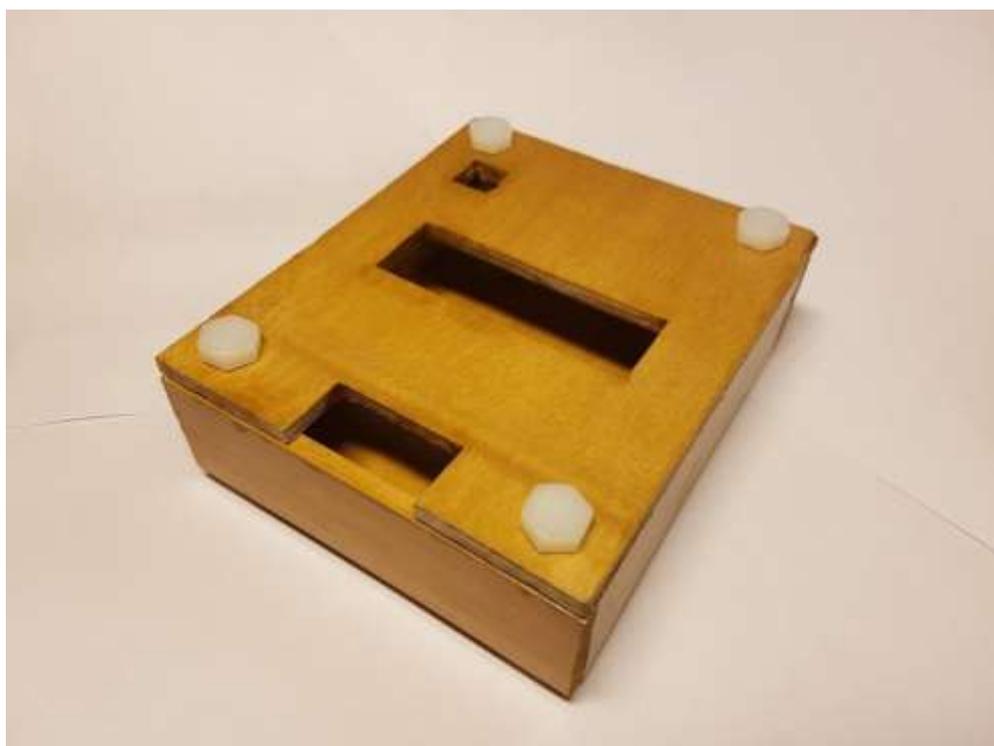


Рисунок 15 – Корпус устройства.

На рисунке 15 показан корпус, имеющий несколько отверстий. Отверстия сверху предназначены для установки дисплея, кнопки управления и сервопривода. Отверстия с торца корпуса необходимы для подключения питания и/или подсоединения устройства к компьютеру для перепрограммирования.

В отладочной плате Arduino Mega и в других элементах устройства для крепления присутствует отверстия, размеры которых соответствуют болтам М3. В корпусе из фанеры такой вариант крепления устройства не вызывает затруднений. На рисунке 16 показано закрепленное и полностью собранное в корпус устройство.



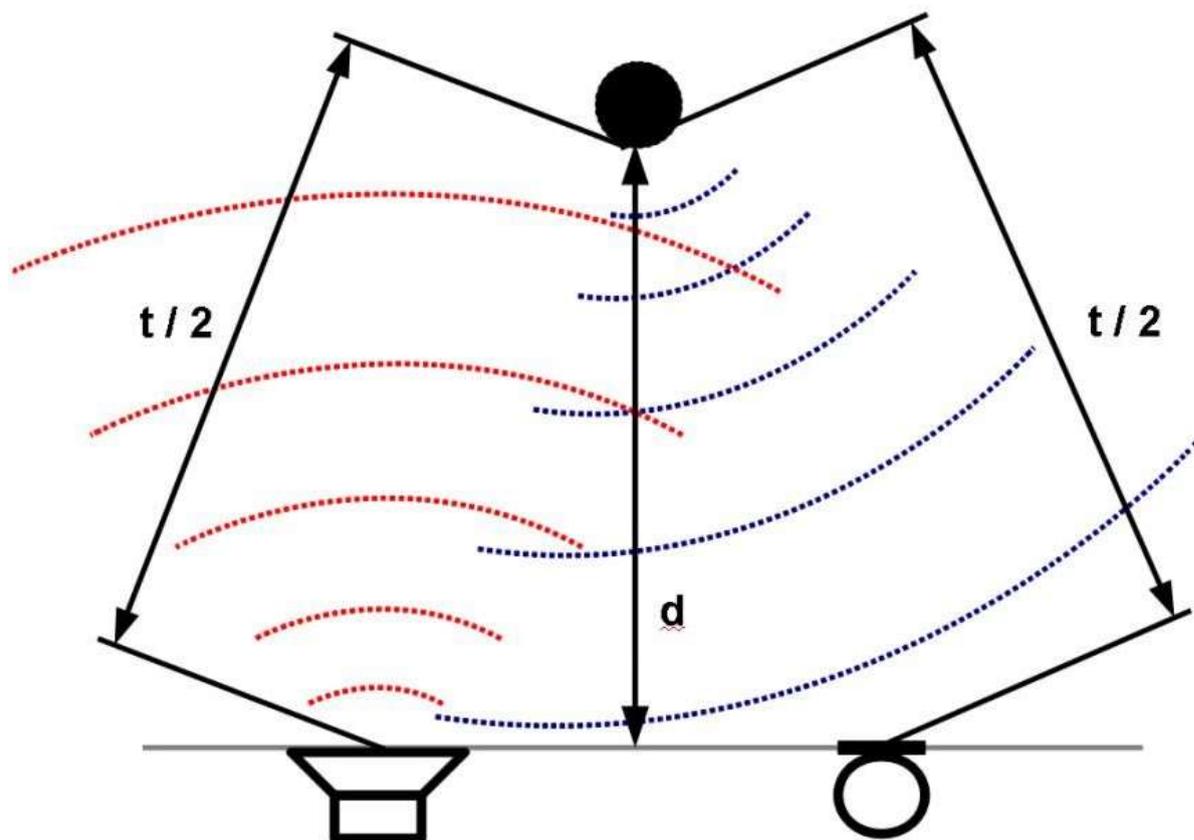
Рисунок 16 – Собранное в корпусе устройство

Итогом всех шагов является устройство измерения расстояния, помещенное в деревянный корпус, как показано на рисунках 15 и 16. Корпус позволяет защитить основные элементы устройства от влаги, пыли и сторонних физических воздействий. Перед использованием устройства следующим шагом необходимо его протестировать и определить основные метрологические характеристики.

3. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ

3.1. Принцип работы устройства измерения расстояния

Разрабатываемое устройство реализовано с помощью ультразвукового датчика, который измеряет расстояние, затем, с помощью аналогово-цифрового преобразования (АЦП) формирует полученную информацию в код. С помощью этого кода программа рассчитывает дистанцию, которая



затем выводится на дисплей. Состоит датчик из двух мембран, одна из которых генерирует направление звуковой волны, а другая принимает отраженное от цели эхо. Звуковой генератор (сенсор) излучает ультразвуковой импульс с некоторым периодом и запускает таймер. Вторая мембрана регистрирует эхо от объекта и останавливает таймер[7]. Принцип работы датчик показан на рисунке 17.

Рисунок 17 – Принцип работы ультразвукового измерителя

Расстояние можно высчитать по времени таймера, который замеряет сколько времени прошло с момента отправки импульса и получения эха и скорости звука в воздухе по формуле:

$$S = \frac{t_{\text{в.у.}} \cdot V_{\text{зв}}}{2}, \quad (1)$$

где S – расстояние от сенсора до объекта, $t_{\text{в.у.}}$ – время, за которое сенсор считывает эхо, $V_{\text{зв}}$ – скорость звука в воздухе.

Таким образом, сенсор принимает сигнал эха и выдает значение расстояния, которое кодируется длительностью электрического сигнала на выходе датчика. Излучение следующего импульса возможно только после считывания эха от предыдущего, по этой причине рекомендованный производителем период между импульсами должен быть не менее 50 мс. В данном проекте на сигнальный порт подается импульс длительностью 10 мкс, после чего модуль излучает 8 пачек ультразвукового сигнала с частотой 40 кГц и обнаруживает их эхо. Принцип действия представлен на рисунке 18. Расстояние до объекта прямо пропорционально ширине эха и рассчитывается по формуле (1).



Рисунок 18 – Принцип действия ультразвукового датчика

Несмотря на свою небольшую стоимость, используемый датчик способен измерять расстояние в диапазоне от 2 до 450 сантиметров с оптимальным углом обзора в 30 градусов, как показано на рисунке 19. Для расширения углов обзора датчик расстояния установлен на сервопривод, который позволяет расширить обзор до 180-ти градусов[8].

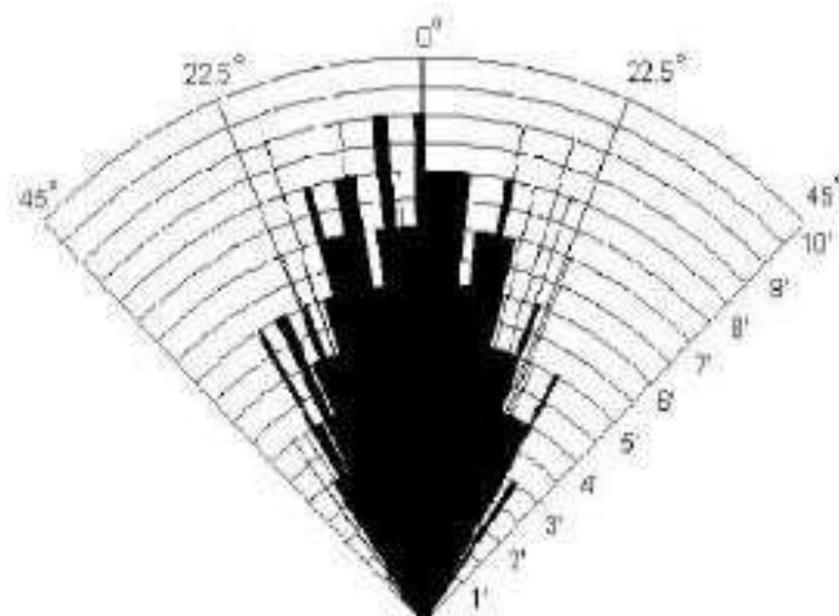


Рисунок 19 – Диаграмма направленности ультразвукового датчика

Для подключения модуля к отладочной плате Arduino на нем присутствует 4 следующих разъема:

- TRIG (T) – вывод выходного сигнала, который посылает импульс до объекта
- ECHO (R) – вывод выходного сигнала, предназначенный для считывания эха от объекта
- VCC - вывод для питания
- GND – земля

Пример подключения датчика к плате показан на рисунке 19

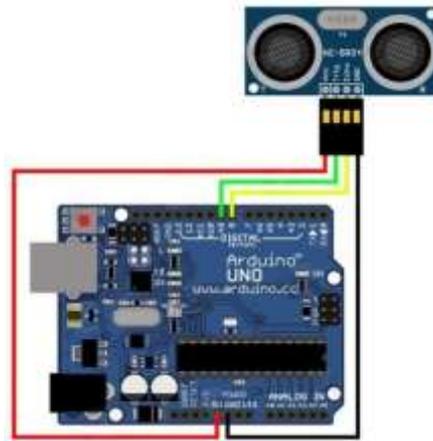


Рисунок 19 – Подключение модуля к Arduino.

После того, как принцип работы модуля изучен, а сам модуль подготовлен к работе, можно создавать код программы. Программа с необходимыми комментариями приведена в приложении В, блок-схема представлена на рисунке 20.

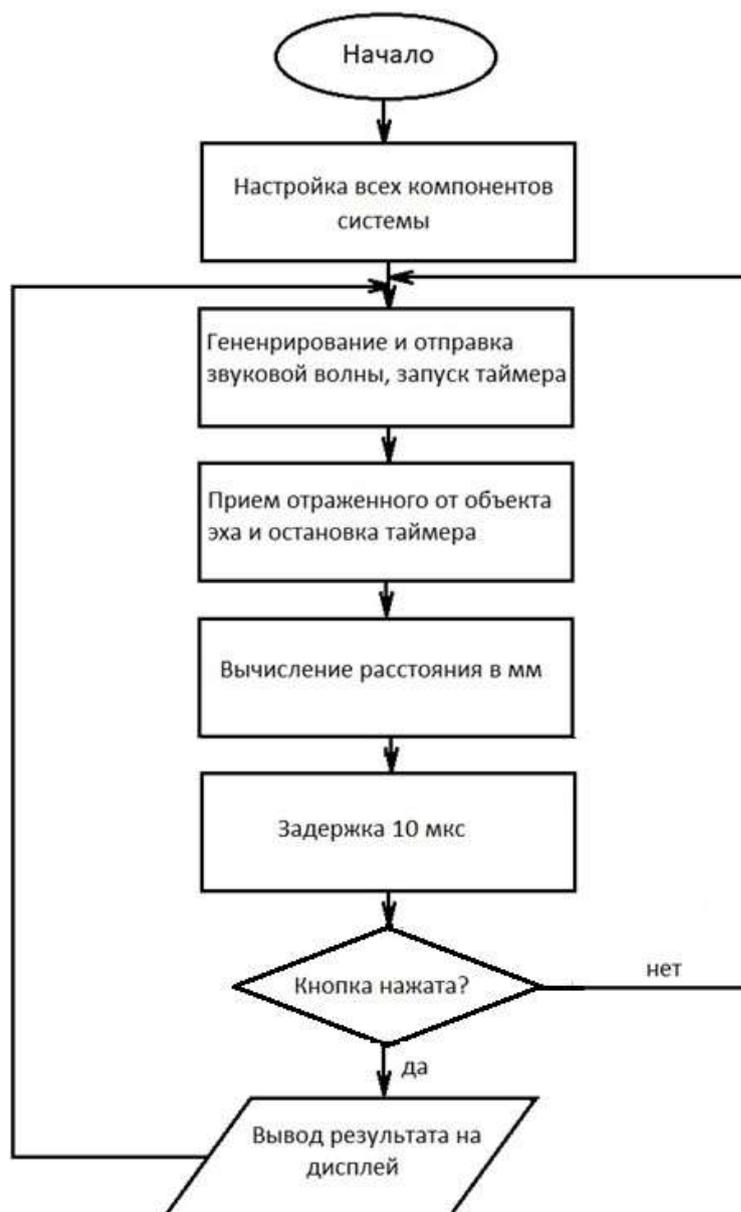


Рисунок 20 – Алгоритм работы программы

Алгоритм программы:

1. Производится настройка всех компонентов системы (активация сервопривода, дисплея и ультразвукового датчика)
2. Генерирование звуковой волны одной из мембран модуля и отправка звуковой волны. Одновременно с этим начинается запуск таймера
3. Считывание эха от объекта, после этой процедуры таймер

останавливается

4. Вычисление расстояния до объекта
5. Задержка в 10 мкс, чтобы прошло эхо от предыдущей операции
6. Проверка состояния кнопки
7. Если кнопка нажата, то вывести результат измерения на дисплей, после чего вернуться к п.2 (повтор пп. 2-8)
8. Если кнопка не нажата, то вернуться к п. 2 (повтор пп. 2-8)

3.2. Сравнение показаний образцового устройства и разработанного.

Для определения характеристик прибора была проведена серия опытов с измерением расстояния образцовым прибором и разработанным устройством в одинаковых условиях. В качестве образцового устройства использовался ультразвуковой дальномер, изображенный на рисунке 21. Данный прибор способен измерять расстояние от 0,5 до 18 метров с точностью $\pm 0.5\%$. Единицы измерения: футы, метры.



Рисунок 21 – Ультразвуковой измеритель расстояния

Разработанный проект является поверяемым устройством. Измерения

проводились на всем возможном диапазоне измерений, то есть, от 2 см до 400 см с шагом 20 см. Результаты измерений приведены в таблице 4.

Таблица 4 – Результаты измерений, полученные разработанным прибором.

№	Показания поверяемого устройства S, см	Показания образцового прибора, см	
		При увеличении S ₀ ув	При уменьшении S ₀ ум
1	20	21	21
2	39	40	40
3	60	61	61
4	78	79	80
5	98	100	101
6	117	119	118
7	136	139	138
8	157	159	158
9	176	178	177
10	197	198	199
11	216	218	219
12	236	239	238
13	257	259	260
14	278	279	280
15	301	302	303
16	319	320	321
17	337	339	338
18	358	359	359
19	376	379	378

20	396	399	399
----	-----	-----	-----

3.3. Расчет основных метрологических характеристик

К основным метрологическим характеристикам относятся: [9]

- Абсолютная погрешность
- Относительная погрешность
- Приведенная погрешность
- Вариация

Для расчета абсолютной погрешности используют формулу разности между показаниями поверяемого устройства и образцового. Из вышесказанного получаем:

$$\Delta S_{ув} = S - S_{0\text{ ув}},$$

$$\Delta S_{ум} = S - S_{0\text{ ум}},$$

ум,

где в соответствии с таблицей 4, S – показания поверяемого устройства, $S_{0\text{ ув}}$ – показания образцового устройства при увеличении показаний по диапaзону, $S_{0\text{ ум}}$ – показания образцового устройства при уменьшении показаний по диапaзону.

В таблице 5 приведены рассчитанные по модулю значения абсолютной погрешности для всего диапaзона измерений.

Таблица 5 – Результаты расчета абсолютной погрешности.

№	Показания поверяемого	Показания образцового прибора, см	Абсолютная погрешность, см
---	-----------------------	-----------------------------------	----------------------------

	устройства S, см	При увели- чении S ₀ ув, см	При умень- шении S ₀ ум, см	При увели- чении ΔS ₀ ув, см	При умень- шении ΔS ₀ ум, см
1	20	21	21	1	1
2	39	40	40	1	1
3	60	61	61	1	1
4	78	79	80	1	2
5	98	100	101	2	3

Продолжение таблицы 5

6	117	119	118	2	1
7	136	139	138	3	2
8	157	159	158	2	2
9	176	178	177	2	1
10	197	198	199	2	2
11	216	218	219	2	3
12	236	239	238	3	2
13	257	259	260	2	3
14	278	279	280	1	2
15	301	302	303	1	2
16	319	320	321	1	2
17	337	339	338	2	1
18	358	359	359	1	1
19	376	379	378	3	2
20	396	399	399	3	3

Для расчета относительной погрешности используется следующая формула:

$$\delta = \frac{\Delta S}{S} \cdot 100\%$$

Если абсолютная погрешность измеряется в единицах измеряемой ве-

личины, то единицами относительной погрешности являются проценты. Для расчета данной погрешности выбирается из каждой пары наибольшее значение абсолютной погрешности по модулю.

Таблица 6 – Результаты расчета относительной погрешности

№	Показания поверяемого устройства S , см	Показания образцового прибора, см		Погрешность, см		
		При увеличении S_0 ув, см	При уменьшении S_0 ум, см	Абсолютная, см		Относительная δ , %
				При увеличении ΔS_0 ув, см	При уменьшении ΔS_0 ум, см	
1	20	21	21	1	1	5
2	39	40	40	1	1	2.6
3	60	61	61	1	1	1.7
4	78	79	80	1	2	2.6
5	98	100	101	2	3	3.1
6	117	119	118	2	1	1.7
7	136	139	138	3	2	2.2
8	157	159	158	2	2	1.3
9	176	178	177	2	1	1.1
10	197	198	199	2	2	1.1
11	216	218	219	2	3	1.4
12	236	239	238	3	2	1.3
13	257	259	260	2	3	1.2
14	278	279	280	1	2	0.6
15	301	302	303	1	2	0.6
16	319	320	321	1	2	0.6
17	337	339	338	2	1	0.6

18	358	359	359	1	1	0.3
19	376	379	378	3	2	0.8
20	396	399	399	3	3	0.8

Для расчета приведенной погрешности используется схожая формула с относительной погрешности, но в ней используется нормируемое значение поверяемого устройства на всем диапазоне измерений S_N вместо показания поверяемого устройства S .

$$\delta = \frac{\delta_{\text{н}}}{S_N} \cdot 100$$

В данном случае нормированное значение будет равно $S_N=400$.

Результаты расчетов приведенной погрешности показаны в таблице 7.

Таблица 7 – Результаты расчетов приведенной погрешности

№	Показание прибора, см	Показания образцового прибора, см		Погрешность, см			
		При увеличении $S_{0\text{ув}}$, см	При уменьшении S_0 ум, см	Абсолютная, см		Относительная δ , %	Приведенная γ , %
				При увеличении $\Delta S_{0\text{ув}}$, см	При уменьшении $\Delta S_{0\text{ум}}$, см		
1	20	21	21	1	1	5	0.25
2	39	40	40	1	1	2.6	0.25
3	60	61	61	1	1	1.7	0.25
4	78	79	80	1	2	2.6	0.5
5	98	100	101	2	3	3.1	0.75
6	117	119	118	2	1	1.7	0.5
7	136	139	138	3	2	2.2	0.75
8	157	159	158	2	2	1.3	0.5
9	176	178	177	2	1	1.1	0.5

Продолжение таблицы 7

10	197	198	199	2	2	1.1	0.5
11	216	218	219	2	3	1.4	0.75

12	236	239	238	3	2	1.3	0.75
13	257	259	260	2	3	1.2	0.75
14	278	279	280	1	2	0.6	0.5
15	301	302	303	1	2	0.6	0.5
16	319	320	321	1	2	0.6	0.5
17	337	339	338	2	1	0.6	0.5
18	358	359	359	1	1	0.3	0.5
19	376	379	378	3	2	0.8	0.5
20	396	399	399	3	3	0.8	0.75

Последним шагом в определении метрологических характеристик для разрабатываемого прибора является вариация. Вариация определяется как «абсолютное значение разности действительных значений измеряемой величины при одном и том же показании прибора, полученном при плавном подходе указателя сначала со стороны меньших, а затем со стороны больших значений»[7]. Формула для расчета вариации следующая:

$$\sigma = \frac{|\Delta S_{0 \text{ ув}} - \Delta S_{0 \text{ ум}}|}{\sigma_{\sigma}} \cdot 100$$

Все результаты расчетов при определении метрологических характеристик разрабатываемого устройства приведены в таблице 8.

Таблица 8 – Результаты расчета основных метрологических характеристик

№	Показания поверяемого устройства S , см	Показания образцового прибора, см		Погрешность, см				
		При увеличении $S_{0ув}$, см	При уменьшении S_0 ум, см	Абсолютная, см		Относительная δ , %	Приведенная γ , %	Вариация H , %
				При увеличении $\Delta S_{0ув}$, см	При уменьшении ΔS_0 ум, см			
1	20	21	21	1	1	5	0.25	0
2	39	40	40	1	1	2.6	0.25	0
3	60	61	61	1	1	1.7	0.25	0
4	78	79	80	1	2	2.6	0.5	0.25
5	98	100	101	2	3	3.1	0.75	0.25
6	117	119	118	2	1	1.7	0.5	0.25
7	136	139	138	3	2	2.2	0.75	0.25
8	157	159	158	2	2	1.3	0.5	0
9	176	178	177	2	1	1.1	0.5	0.25
10	197	198	199	2	2	1.1	0.5	0
11	216	218	219	2	3	1.4	0.75	0.25
12	236	239	238	3	2	1.3	0.75	0.25
13	257	259	260	2	3	1.2	0.75	0.25
14	278	279	280	1	2	0.6	0.5	0.25
15	301	302	303	1	2	0.6	0.5	0.25
16	319	320	321	1	2	0.6	0.5	0.25
17	337	339	338	2	1	0.6	0.5	0.25
18	358	359	359	1	1	0.3	0.5	0

19	376	379	378	3	2	0.8	0.5	0.25
20	396	399	399	3	3	0.8	0.75	0

На рисунках 22 и 23 изображены показания поверяемого и образцового приборов с увеличением расстояния до объекта.

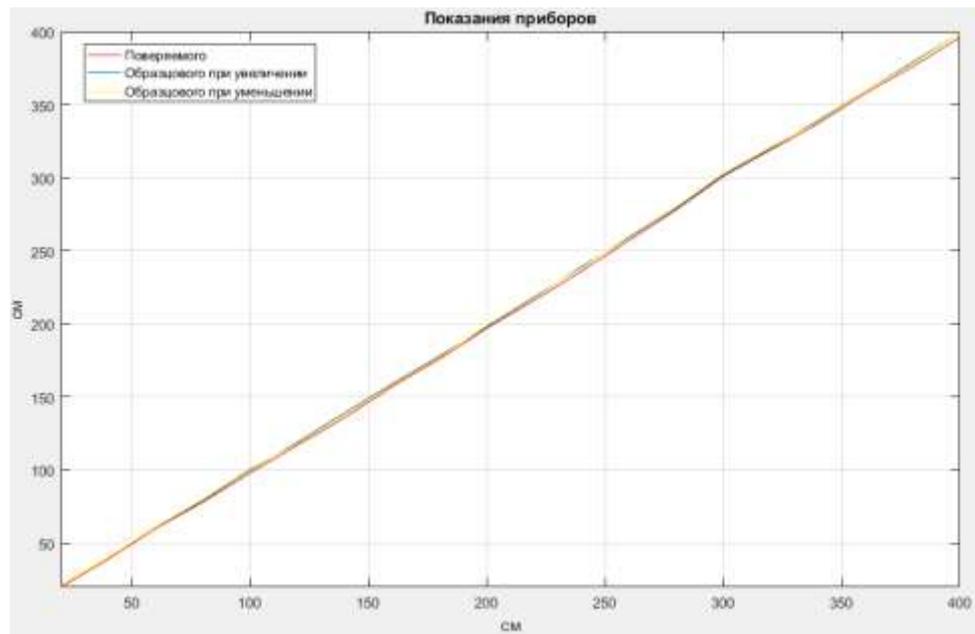


Рисунок 22 – Показания поверяемого и образцового устройств на всем диапазоне измерений

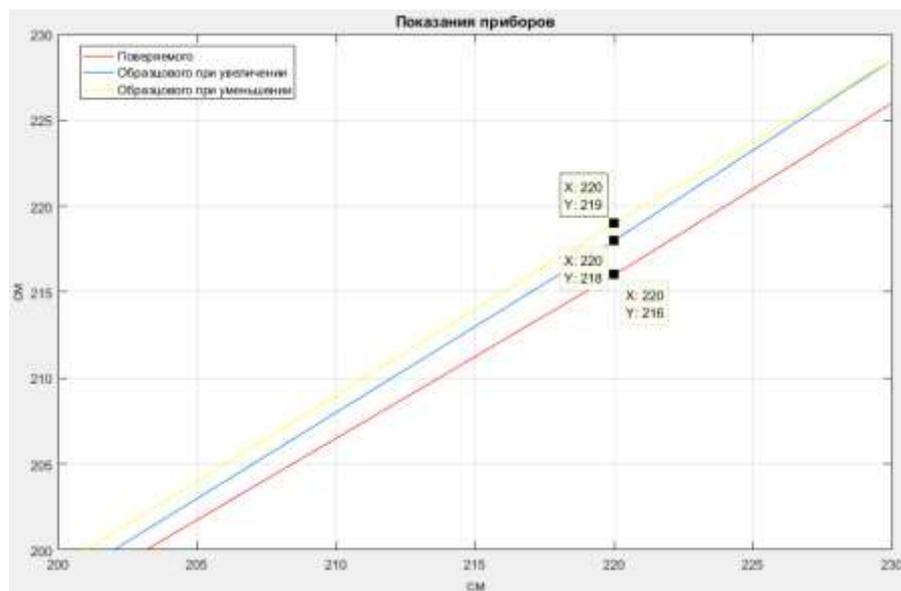


Рисунок 23 – Показания поверяемого и образцового устройств на диапазоне от 200 до 230 сантиметров

Максимальная абсолютная погрешность при измерении расстояния составила 3 сантиметра в выбранном диапазоне, а значит, результат измерений можно представить в следующем виде:

$$S_p = S \pm 3 \text{ см},$$

где S_p – значение расстояния в данный момент времени, а S – значение расстояния, показываемое устройством на дисплее после его вычисления. В таблице 8 приведены результаты расчета основным метрологических характеристик разработанного устройства:

- Относительная погрешность: $\delta=5\%$ при $S=20$ см.
- Приведенная погрешность: $\gamma=0.75\%$
- Вариация: $H=0.25\%$

Устройство может использоваться для измерения расстояний от 2 сантиметров до 400 сантиметров, так как при измерении расстояния меньше двух сантиметров эхо от предыдущего считывания перебивает последующее, а при измерении более чем 400 сантиметров, плата будет ждать отраженный

сигнал, который не поступит на сенсор, считывающий эхо. Так как, зачастую, для измерения расстояния менее 20 сантиметров существует возможность воспользоваться другими методами измерения (линейка, рулетка), то оптимальным решением будет выбрать именно это значения за начало диапазона. Для измерения объектов на удалении более 400 сантиметров есть уже смысл воспользоваться лазерными измерителями, которые более точны на дальних расстояниях.

ЗАКЛЮЧЕНИЕ

Итогом работы стало смоделированное и собранное устройство измерения расстояния на базе программируемой отладочной платы Arduino Mega 2560. Устройство имеет рабочий диапазон измерений от 20 см до 400 см и способно проводить измерение расстояния до объектов средних и крупных размеров, а результат измерения имеет абсолютную погрешность 3 сантиметра. Прибор с такими характеристиками можно использовать в широком диапазоне задач. Невосприимчивость прибора к более мелким объектам можно объяснить частотой волны в 40 кГц, для повышения чувствительности соответственно необходимо повысить частоту излучаемого сигнала. Погрешность в несколько сантиметров объясняется присутствием частичных отражений – паразитных эхо-сигналов, которые искажают получаемые результат. Как правило, в роли таких паразитных эхо-сигналов могут выступать криволинейные по отношению к направлению сигнала поверхности.

Перед выбором платы Arduino Mega 2560 был изучен весь модельный ряд этого производителя и несколько плат других марок. Выбранная для прибора плата является оптимальным вариантом, так как содержит в себе необходимые для создания прибора характеристики, а также имеет запас мощности для последующих модификаций устройства. Далее была подобрана элементная база устройства, подключены и откалиброваны различные модули для корректной работы в единой системе.

Работа над ультразвуковым измерителем состояла из нескольких этапов. Создание идеи на бумаге и моделирование проекта, после чего сборка функционирующего прототипа. После всех проверок и тестирований временные соединения элементов были заменены на постоянные, а для устройства был спроектирован корпус. Основное назначение корпуса – защита прибора от попадания пыли и влаги, а так же любых других физических воздействий.

Для того чтобы запрограммировать измеритель расстояния использовался язык, основанный на C++ в среде Arduino IDE.

Завершающим шагом в работе было определение основных метрологических характеристик прибора: абсолютной погрешности, относительной погрешности, приведенной погрешности и вариации.

Во время выполнения выпускной квалификационной работы были применены навыки, полученные в процессе изучения таких предметов как:

- Основы теории сигналов
- Основы проектирования приборов и систем
- Метрология
- Цифровые и аналоговые измерительные устройства
- Микропроцессорные устройства в информационно-измерительной технике

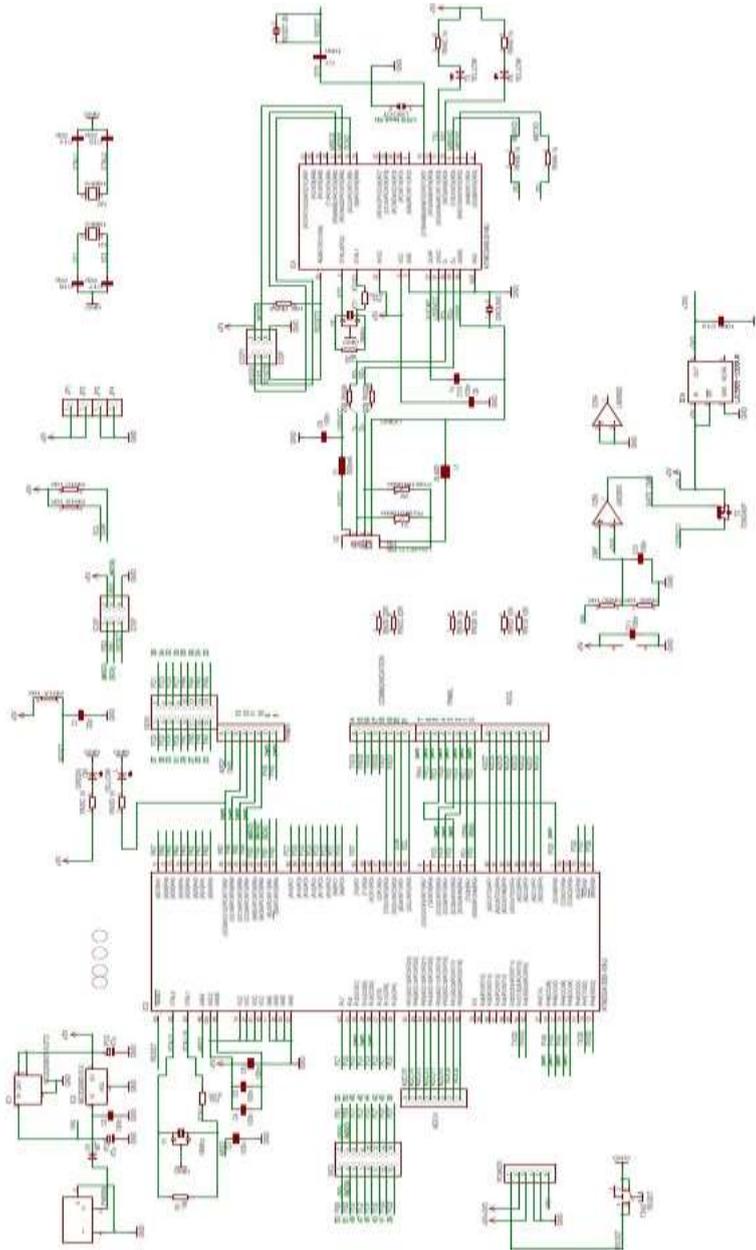
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Аппаратная платформа Arduino [Электронный ресурс]. URL: <https://www.arduino.ru/> (дата обращения 25.04.2017).
2. Соммер У. Программирование микроконтроллерных плат Arduino/*Freeduino*. - СПб.: БХВ-Петербург, 2012. -256 с., ил. - (Электроника).
3. Хабрахабр/Недостатки javascript [Электронный ресурс]. URL: <https://habrababr.ru/sandbox/70666> (дата обращения 26.04.2017).
4. Байда А. С. Использование платформы Arduino при подготовке специалистов автомобильной отрасли // Научно-методический электронный журнал «Концепт». – 2016. – № 5 (май). – 0,4 п. л.
5. Хоровиц П., Хилл У. Искусство схемотехники: Пер. с англ. — Изд. 2-е. — М.: Издательство БИНОМ. — 2016. —704 с., ил.
6. Инженерная и компьютерная графика: учеб. пособие / В. П. Большаков, В. Т. Тозик, А. В. Чагина. – СПб.:БХВ-Петербург, 2013. – 288 с.: ил.
7. Микроконтроллеры AVR семейств Tiny и Mega фирмы ATMEL, 5-е изд., стер. – М.: Издательский дом «Додэка-XXI», 2008. – 560с.
8. Шонфелдер Г., Шнайдер К. Измерительные устройства на базе микропроцессоров *ATmega*. - СПб.: БХВ-Петербург, 2012. -288 с., ил. - (Электроника).
9. Метрология, стандартизация и сертификация: Методические указания к лабораторным работам / под ред. Б. Я. Авдеева. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2008 г.
10. ГОСТ Р 50923-96 Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения. – Введ. 1996-07-10. — М.: Изд-во стандартов, 1996. — 9 с. (дата обращения: 18.05.2017)

ПРИЛОЖЕНИЕ А

Arduino™ Mega 2560 Reference Design

© 2013 Intel Corporation. All rights reserved. Intel, the Intel logo, and the Intel logo with "Intel Inside" are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. Intel, the Intel logo, and the Intel logo with "Intel Inside" are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. Intel, the Intel logo, and the Intel logo with "Intel Inside" are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.



ПРИЛОЖЕНИЕ В

```
#include <Servo.h>.
```

```

#include <LiquidCrystal_I2C.h>
#include <CyberLib.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,8,2);    // Устанавливаем дисплей

//входы портов для эхолота const
int echoPin = 11;
const int pin = 8;

double duration; //указания типов переменных
double distance;
Servo myServo; // создание переменной сервопривода

void setup() {
  pinMode(pin, INPUT);
  lcd.init();
  lcd.backlight();
  D10_Out;
  D11_In;
  myServo.attach(12); // порт для сервопривода
}

double calculateDistance() //функция расчета дистанции
{

  D10_Low; delay_ms(2);
  D10_High;
  delay_ms(10);
  D10_Low;

```

```

duration = pulseIn(echoPin, HIGH); distance=
duration*0.034/2;
return distance;
}

```

```

void loop() {

    //перемещение сервопривода с 165 до 15 градусов {
    for(int i=15;i<=165;i++)
    {
        if (D8_Read == LOW)
        {delay(1500);
            break;
        }
        myServo.write(i); delay_us(60);
        distance = calculateDistance();
        lcd.setCursor(0,0);
        lcd.print("Dist(cm):");
        lcd.print(distance); lcd.print("
            ");

    }
    // перемещение сервопривода с 165 до 15 градусов for(int i =
    165; i>15 ;i--)
    {
        if (D8_Read == LOW)
        { delay(1500);
            break;

```

```
    }  
    myServo.write(i); delay_us(60);  
    distance = calculateDistance();  
    lcd.setCursor(0,0);  
    lcd.print("Dist(cm):");  
    lcd.print(distance); lcd.print("  
                ");  
    }  
}
```