

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт машиностроения

(наименование института полностью)

Кафедра «Промышленная электроника»

(наименование)

11.03.04 Электроника и микроэлектроника

(код и наименование направления подготовки / специальность)

Электроника и робототехника

(направленность (профиль) / специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему Электронная система помощи при парковке автомобиля в гараж

Обучающийся

Д.А. Чернышов

(Инициалы Фамилия)

(личная подпись)

Руководитель

И. И. Золотов

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

О. А. Головач

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Название дипломной работы: «Электронная система помощи при парковке автомобиля в гараж».

Выпускная работа состоит из введения, пяти разделов, заключения, двадцати шести рисунков, списка литературы, включая зарубежные источники, графической части на пяти листах формата А1.

В данной работе представлен процесс проектирования и создания модели системы помощи при парковке автомобиля в гараж. Особенностью данной разработки является гибкость применения в зависимости от планировки гаража.

Целью работы является разработка и проектирование системы помощи при парковке в гараж, а также написание программы для этой системы.

Бакалаврская работа может быть разделена на следующие логически взаимосвязанные части: анализ состояния вопроса, разработка и обоснование схмотехнического решения, разработка управляющей программы устройства, проектирование опытного образца устройства. В анализе вопроса рассмотрены похожие решения и оценены их достоинства и недостатки. В следующем разделе рассмотрели возможные электронные устройства нашей системы, а также произвели выбор устройств. В разделе о разработке управляющей программы произвели описание кода программы. В конце произвели моделирование устройства.

Подводя итоги, хотелось бы отметить, что в мире все больше развивается автоматизация жизни человека. Все это необходимо для упрощения жизни и повышения безопасности.

ABSTRACT

Title of the thesis: «An electronic assistance system for parking a car in a garage. »

The final work consists of an introduction, four sections, a conclusion, twenty illustrations, a list of references including foreign sources, and a graphic part on five A1-sized sheets.

This paper presents the design and development of a car parking assistance system for a garage. The special feature of this project is its flexibility, allowing it to be applied in different garage layouts.

The aim of the study is to develop and design a system that can assist with parking in a garage, as well as to write a program to control the system.

The bachelor's project can be divided into several logically connected parts: an analysis of the current situation, the development and justification of circuit design solutions, the creation of a control program for the device, and the design of a prototype. In the analysis of the issue, various solutions were considered and their pros and cons were evaluated. In the next section, we looked at the possible electronic components of our system and made a selection. In the development of the control software, a description of the code was written. Finally, a simulation of the device was carried out.

To summarize, I would like to point out that the automation of human life is increasingly becoming a reality in the world. This is necessary in order to simplify life and enhance safety.

Содержание

Введение.....	5
1 Анализ состояния вопроса	6
1.1 Анализ известных решений	6
1.2 Формулировка задач исследования.....	10
2 Разработка и обоснование схемотехнического решения	12
2.1 Обоснование выбора микроконтроллера	12
2.2 Обоснование выбора ультразвукового датчика.....	17
2.3 Обоснования выбора светодиодной ленты	22
2.4 Разработка структурной схемы	24
2.5 Разработка принципиальной схемы	25
3 Разработка алгоритма работы устройства	27
4 Разработка управляющей программы устройства.....	33
4.1 Подключение библиотек и объявление переменных	33
4.2 Программирование ультразвуковых датчиков	35
4.3 Программирование логики и светодиодной ленты	36
5 Проектирование модели устройства и отладка в пакете автоматизированного проектирования	40
Заключение	45
Список используемой литературы	46
Приложение А Программный код алгоритма работы.....	48

Введение

В современном мире одной из основных тенденций является стремление к повышению комфорта и безопасности жизни, а также снижению сложности быта. Не обошла данная тенденция и автомобильную сферу.

В наше время почти у каждой семьи есть личный автомобиль, а иногда даже не один. Наличие необходимости хранения транспортных средств и желание владельцев иметь гарантии защиты автомобиля от воздействия окружающей среды и, в некоторой степени, от угона, приводит к появлению почти у каждой семьи гаража в том или ином виде. Однако наличие гаража, зачастую имеющего небольшие габаритные размеры, порождает усложняющий быт человека элемент – процесс парковки в ограниченном пространстве. Ошибки в процессе парковки в усложненных условиях особенно часто приводят к повреждению автомобиля, ремонт которого в наше время либо дорогостоящий, либо крайне осложнен, ввиду логистических трудностей с поставкой деталей.

Снизить риск повреждения кузовных элементов автомобиля при парковке в ограниченном пространстве помогают электронные ассистенты. На каких-то моделях встречается штатные системы помощи при парковке, а на других автовладелец самостоятельно, либо с помощью специалистов, устанавливает нештатные системы. Однако в нашей действительности, с уходом с официального рынка именитых производителей, возникает проблема с выбором надежного, качественного и относительно недорогого электронного устройства. Зачастую представленные на рынке системы оказываются ненадежными, притом, что имеют высокую стоимость. Соответственно возникает необходимость повышения надежности таких устройств, с сохранением невысокой стоимости готового устройства. Таким образом, целью настоящей выпускной квалификационной работы является повышение надежности электронных систем помощи при парковке, с сохранением невысокой стоимости готового устройства.

1 Анализ состояния вопроса

1.1 Анализ известных решений

Существует несколько решений для помощи парковки в гараж. Рассмотрим некоторые из них:

1. Park ranger parking aid



Рисунок 1 – Park ranger parking aid

Данная система довольно проста в использовании. Для ее работы необходимо поставить автомобиль в гараж и наклеить на боковую стену это устройство. Красные полосы этого устройства должны совпасть вертикально. В следующий раз при парковке двигайтесь до того момента как линии также совпадут. Крайне зависит от положения сидения.

Плюсы:

– Низкая стоимость,

– Простота монтажа;

Минусы:

– Низкая точность и как следствие ненадежность,

– Отсутствие возможности быстрой регулировки.

2. Parking target IPI-100 peel-n-stick parking aid



Рисунок 2 – Parking target IPI-100 peel-n-stick parking aid

Это устройство также довольно простое в использовании. Для его использования необходимо поставить машину в гараж после чего приклеить на пол под колесо.

Плюсы:

– Экономичное устройство,

– Простота установки;

Минусы:

- Подвижность, а, следовательно, ненадежность, а также,
- Низкая точность при парковке,
- Отсутствие возможности быстрой регулировки.

3. Maxsa 37312 dual laser parking assist



Рисунок 3 – Maxsa 37312 dual laser parking assist

Maxsa 37312 работает за счет проецирования двух лазерных лучей на пол гаража. Лазеры можно регулировать в зависимости от того места, где вам удобно их заметить, к примеру на приборную панель. Для определения позиции необходимо опять же сначала поставить автомобиль, а потом отрегулировать лазер.

Плюсы:

- Регулировка лазера для точной парковки,
- Имеется датчик движения для активации работы,
- Подходит для больших гаражей;

Минусы:

- Высокая стоимость,
- Отсутствие гибкости системы.

4. Cobra tuning LED stop sign



Рисунок 4 – Cobra tuning LED stop sign

Данное устройство предназначено для обозначения места остановки автомобиля. Но, прежде чем устройство работало по назначению необходимо выставить автомобиль в гараже после чего поставить знак перед транспортным средством.

Плюсы:

- Простота установки,
- Простота в использовании;

Минусы:

- Высокая стоимость,
- Подвижность,
- Неточность.

1.2 Формулировка задач исследования

Анализируя предложенные на рынке устройства помощи при парковке, были выявлены недостатки данной группы устройств, в зависимости от их типа. Для технических решений, механического типа, имеющих низкую стоимость характерны такие недостатки как низкая точность и надежность системы, а также отсутствие возможности настройки устройства под определенные внешние условия без изменения или переработки конструкции, иными словами – отсутствие гибкости. Для технических решений, имеющих, согласно опыту эксплуатации пользователями, достаточную, при определенных условиях, надежность и точность характерна высокая стоимость готового устройства и достаточно ограниченная адаптивность системы под конкретные внешние условия

В рамках настоящей выпускной квалификационной работы предлагается для устранения недостатков известных технических решений, а, следовательно, достижения поставленной цели разработать электронное устройство помощи при парковке в гараж.

Таким образом, можно выделить следующие задачи выпускной квалификационной работы:

- Разработать схемотехническое решение электронной системы помощи при парковке автомобиля в гараж;

- Разработать алгоритм работы устройства;
- Разработать управляющую программу, реализующую заданный алгоритм работы;
- Провести тестирование разработанной модели устройства в программе автоматизированного проектирования.

Выводы по разделу

В ходе анализа рынка известных решений были выявлены характерные недостатки, присущие данной группе устройств:

- Для механических систем – низкая точность и надежность, а также, зачастую, отсутствие адаптивности;
- Для электрических систем – высокая стоимость, а также, достаточно ограниченная адаптивность;

Для достижения поставленной цели предложено использовать электронное устройство помощи при парковке в гараж, разработка которого приводит к необходимости решения следующих задач:

- Разработать схемотехническое решение электронной системы помощи при парковке автомобиля в гараж;
- Разработать алгоритм работы устройства;
- Разработать управляющую программу, реализующую заданный алгоритм работы;
- Провести тестирование разработанной модели устройства в программе автоматизированного проектирования.

2 Разработка и обоснование схмотехнического решения

2.1 Обоснование выбора микроконтроллера

Для реализации данной системы необходимо выбрать микроконтроллер. По исходным данным напряжение питания системы управления, а также напряжение цепей управления 5 Вольт постоянного тока. В качестве системы управления использовать микроконтроллер. Эксплуатация устройства – в помещении.

Выбирать будем между Arduino Nano и Arduino UNO [15].

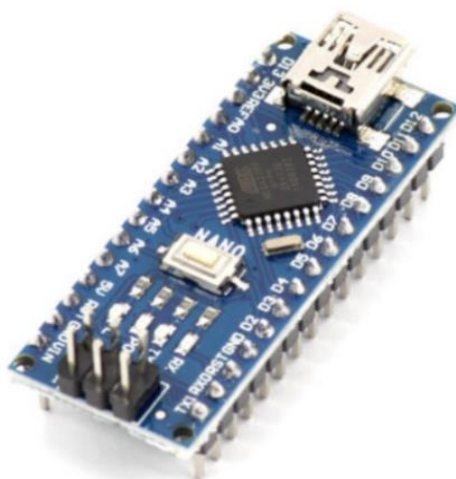


Рисунок 5 – Arduino Nano

Arduino Nano оснащена микроконтроллером ATmega328, который имеет тактовую частоту 16 МГц и в зависимости от чипа от 16 до 32 Кбайт памяти Flash, от 1 до 2 Кбайт оперативной памяти и от 512 до 1 Кбайт энергонезависимой памяти EEPROM [12].

Одно из различий между Arduino Nano и Arduino UNO заключается в том, что у Nano нет отдельного разъёма для подключения внешнего источника питания. Вместо этого для питания и связи с компьютером используется кабель mini-USB. Также у Nano есть штырьковые контакты, благодаря которым плату можно установить на макетную плату.

Одно из главных достоинств Arduino Nano — её компактные размеры. Это позволяет использовать её в проектах, где требуется небольшая плата. Кроме того, Arduino Nano — более доступная альтернатива Arduino UNO, что делает её привлекательной для широкого круга пользователей. В итоге Arduino Nano — отличная платформа для реализации разнообразных проектов, требующих компактности и надёжности. Она обладает всеми функциями и возможностями Arduino UNO, но при этом занимает меньше места.



Рисунок 6 – Arduino UNO

Плата основана на чипе ATmega328, этот чип оснащён 32 килобайтами флеш-памяти, 2 килобайтами SRAM и 1 килобайтом EEPROM. На его периферии находятся 14 дискретных каналов ввода и вывода и 6 аналоговых каналов, а также два вывода с аппаратным прерыванием [11, 7]. Благодаря этим характеристикам контроллер способен решать основные задачи. Также следует подчеркнуть полезность этого микроконтроллера в сфере обучения и проверки функциональности. Одно из ключевых преимуществ данной платы заключается в том, что она является одной из самых доступных и распространённых. Благодаря этому её легко найти и приобрести. Эта плата отлично подходит для освоения работы с микроконтроллерами, однако на ней можно успешно реализовывать и более сложные проекты, не требующие множества цифровых входов.

Основной плюс и причина выбора данной платы стал разъем для внешнего питания платы Power Jack (рисунок 7).

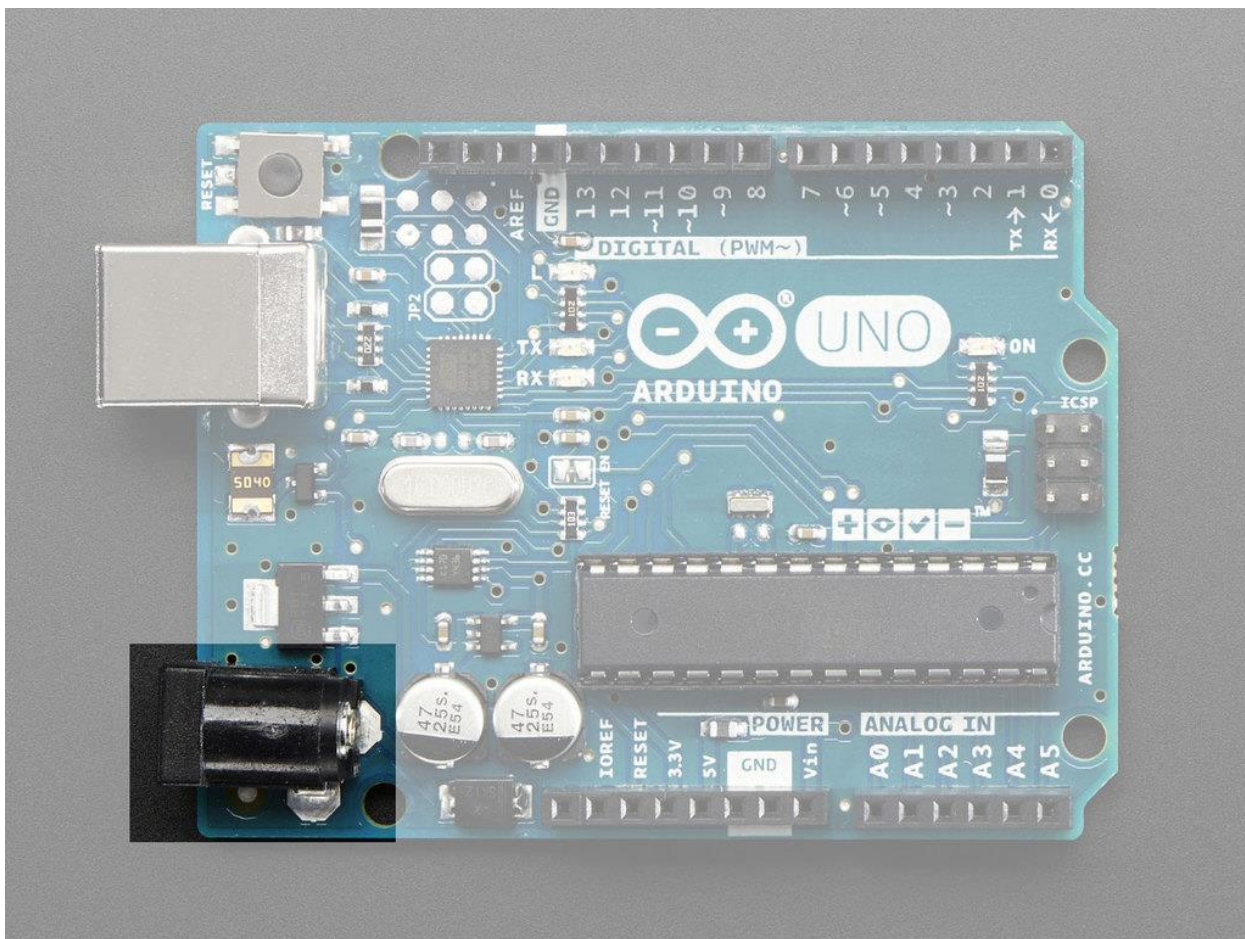


Рисунок 7 – DC Power Jack

Данный разъем рассчитан на постоянный ток от 6 до 20 Вольт. Но рекомендуемое напряжение составляет 9 Вольт [4, 5]. Тогда возникает вопрос почему рекомендуемое напряжение 9 Вольт, если рабочее напряжение составляет 5 В. При создании данной платы был продуман этот вопрос. Arduino имеет встроенный регулятор напряжения (рисунок 8).

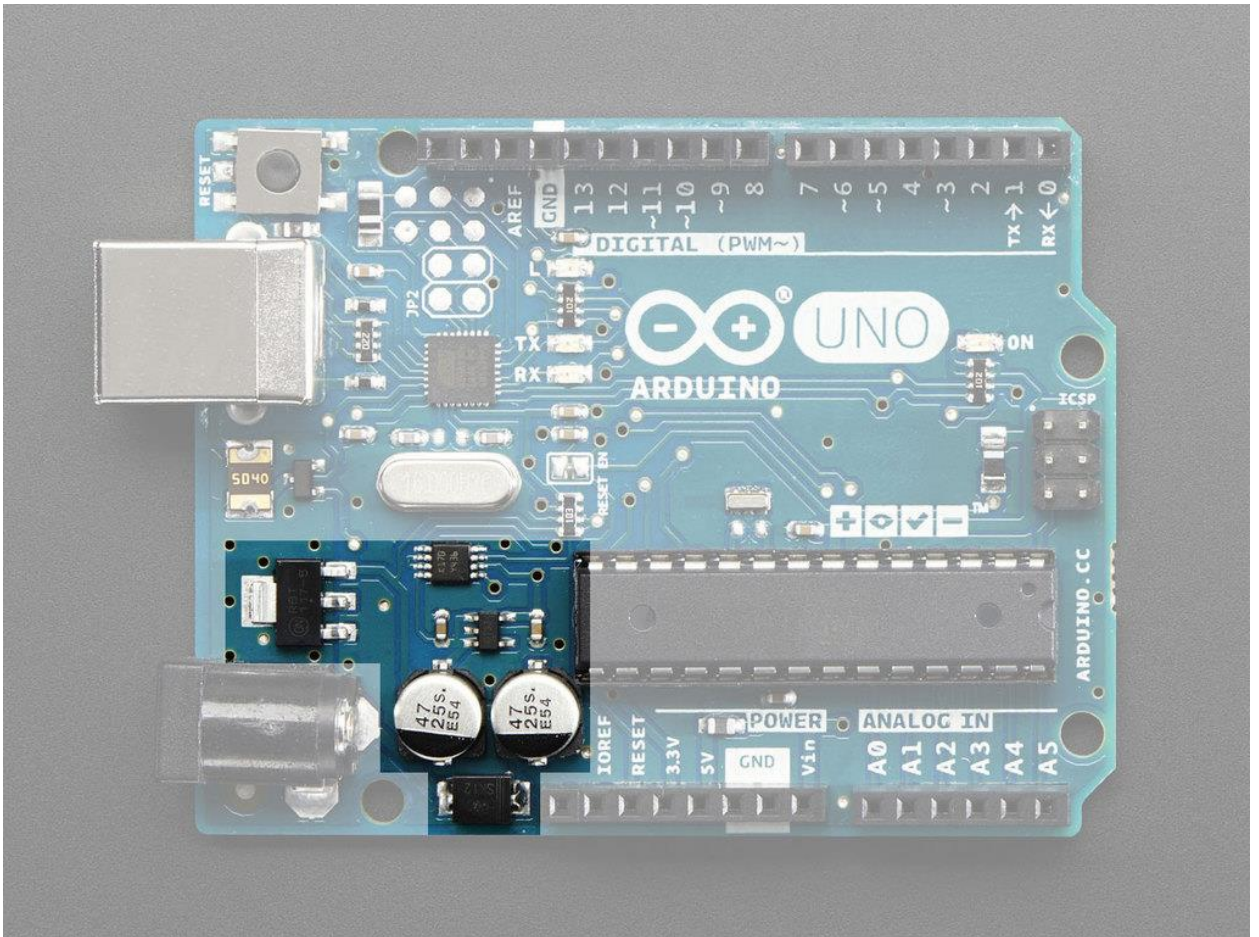


Рисунок 8 – Регулятор напряжения

SPX1117M3-L-5 Regulator [10, 20] данный регулятор установлен на плате. Он позволяет подавать на плату до 20 Вольт положительной полярности. Также есть ограничение по силе тока до 0,8 Ампер, чего отлично хватит для нашего устройства. Следует продумать выбор блока питания для платы, чтобы регулятор долго прослужил, а вместе с ним и плата. Для стабильной работы платы понадобится 9 Вольт, чтобы на выходе стабилизатора были стабильные 5 Вольт. При подаче на плату 5 Вольт через разъем Power Jack, регулятор срежет это напряжение исходя из datasheet (рисунок 9) примерно на 1,1 Вольта и в результате на выходе будет примерно 3,9 Вольта. Чего абсолютно не хватит для работы микроконтроллера, да и всей системы в целом.

ELECTRICAL SPECIFICATIONS

Specifications with standard type are for an Operating Ambient Temperature of $T_A = 25^\circ\text{C}$ only; limits applying over the full Operating Junction Temperature range are denoted by a "*". Minimum and Maximum limits are guaranteed through test, design, or statistical correlation. Typical values represent the most likely parametric norm at $T_J = 25^\circ\text{C}$, and are provided for reference purposes only. Unless otherwise indicated, $C_{IN} = C_{OUT} = 10\mu\text{F}$, $T_A = 25^\circ\text{C}$.

Parameter	Min.	Typ.	Max.	Units	Conditions
1.5V Version					
Output Voltage	1.485	1.500	1.515	V	$I_{OUT}=5\text{mA}$, $V_{IN}=3.0\text{V}$, $T_J=25^\circ\text{C}$
	1.470		1.530		• $5\text{mA} \leq I_{OUT} \leq 800\text{mA}$, $2.9\text{V} \leq V_{IN} \leq 10\text{V}$
1.8V Version					
Output Voltage	1.782	1.800	1.818	V	$I_{OUT}=5\text{mA}$, $V_{IN}=3.3\text{V}$, $T_J=25^\circ\text{C}$
	1.764		1.836		• $5\text{mA} \leq I_{OUT} \leq 800\text{mA}$, $3.2\text{V} \leq V_{IN} \leq 10\text{V}$
2.5V Version					
Output Voltage	2.475	2.500	2.525	V	$I_{OUT}=5\text{mA}$, $V_{IN}=4.0\text{V}$, $T_J=25^\circ\text{C}$
	2.450		2.550		• $5\text{mA} \leq I_{OUT} \leq 800\text{mA}$, $3.9\text{V} \leq V_{IN} \leq 10\text{V}$
3.3V Version					
Output Voltage	3.267	3.300	3.333	V	$I_{OUT}=5\text{mA}$, $V_{IN}=4.8\text{V}$, $T_J=25^\circ\text{C}$
	3.234		3.366		•
5.0V Version					
Output Voltage	4.950	5.000	5.050	V	$I_{OUT}=5\text{mA}$, $V_{IN}=6.5\text{V}$, $T_J=25^\circ\text{C}$
	4.900		5.100		• $5\text{mA} \leq I_{OUT} \leq 800\text{mA}$, $6.4\text{V} \leq V_{IN} \leq 12\text{V}$
All Voltage Options					
Reference Voltage	1.238	1.250	1.262	V	$I_{OUT}=5\text{mA}$, $(V_{IN} - V_{OUT})=2\text{V}$, $T_J=25^\circ\text{C}$
	1.225		1.270		• $5\text{mA} \leq I_{OUT} \leq 800\text{mA}$, $1.4\text{V} \leq (V_{IN}-V_{OUT}) \leq 10\text{V}$
Output Voltage Temperature Stability		0.3		%	
Line Regulation (note 1)		3	7	mV	$V_{INMIN} \leq V_{IN} \leq 12\text{V}$, $V_{OUT} = \text{Fixed/Adj.}$, $I_{OUT}=5\text{mA}$
Load Regulation (note 1)		6	12	mV	• $5\text{mA} \leq I_{OUT} \leq 800\text{mA}$, $V_{OUT} = \text{Fixed/Adj.}$
Dropout Voltage (note 2)		1.00	1.20	V	• $I_{OUT}=100\text{mA}$
		1.05	1.25		• $I_{OUT}=500\text{mA}$
		1.10	1.30		• $I_{OUT}=800\text{mA}$
Quiescent Current		5	10	mA	• $4.25\text{V} \leq V_{IN} \leq 6.5\text{V}$
Adjust Pin Current		50	120	μA	•
Current Limit	1.0	1.5	2.0	A	$(V_{IN} - V_{OUT})=5\text{V}$
Thermal Regulation		0.01	0.1	%/W	25°C , 30ms pulse
Ripple Rejection	60	75		dB	$f_{\text{RIPPLE}}=120\text{Hz}$, $(V_{IN} - V_{OUT})=2\text{V}$, $V_{\text{RIPPLE}}=1\text{V}_{\text{PP}}$

Рисунок 9 – Спецификация регулятора



Рисунок 10 – Блок питания

На рисунке представлен пример блока питания для данного проекта. Блок имеет встроенную регулировку напряжения. Регулятор выставлен на 9 Вольт.

2.2 Обоснование выбора ультразвукового датчика

Исходя из заданных данных, что система будет находиться в помещении выбор будет производиться из двух датчиков, а именно GP2Y0A021YK0F и HC-SR04.

GP2Y0A021YK0F это инфракрасный дальномер (рисунок 11), принцип его работы довольно прост датчик излучает луч света в инфракрасном спектре, который отражается от поверхности объекта и возвращается, обратно попадая на приемник.

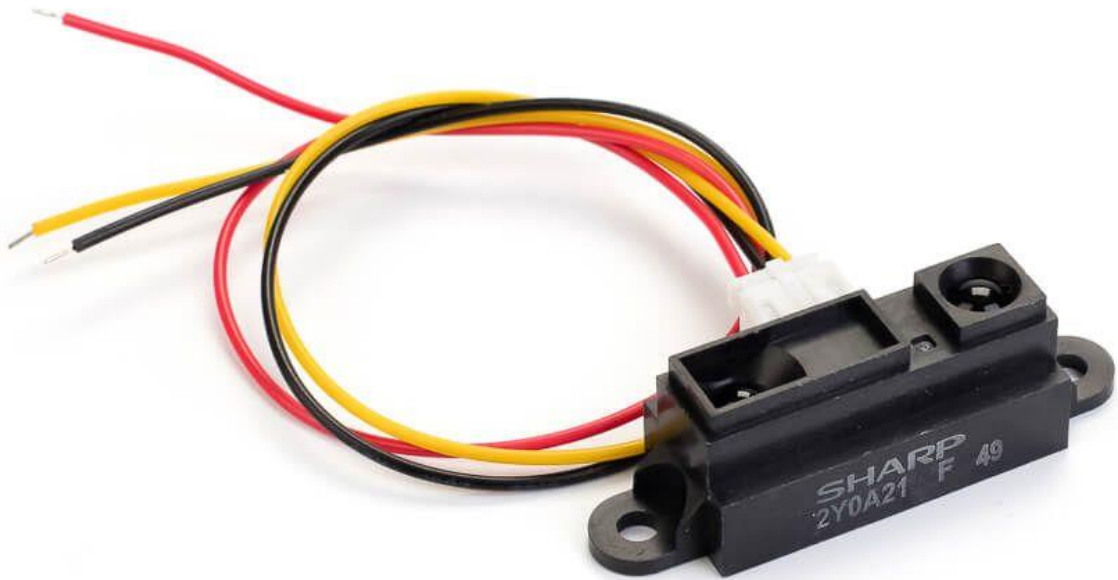


Рисунок 11 – GP2Y0A021YK0F инфракрасный дальномер

Технический характеристики:

- Диапазон измерения 10–80 сантиметров,
- Напряжения питания 4,5–5 Вольт,
- Потребляемый ток 30–40 миллиампер,
- Угол обзора 25°.

Данный датчик через мерно подвержен воздействию солнца, которые могут вывести его в не рабочее состояние [13]. Также из недостатков использование датчика невозможно со светопоглощающими объектами.

HC-SR04 это ультразвуковой датчик представленный на рисунке 12.



Рисунок 12 – Ультразвуковой датчик HC-SR04

Этот датчик работает по принципу сонара: он излучает пучок ультразвука и измеряет время, которое требуется для возвращения отраженного от объекта сигнала. Этот временной интервал служит показателем расстояния до объекта путем пересчета по формуле времени в расстояние.

Датчик обладает высокой точностью измерения и имеет высокую надежность. Диапазон измерения дальности составляет от 2 до 400 сантиметров. Работа устройства не подвержена значительному влиянию электромагнитных излучений и солнечной энергии.

Технические параметры HC-SR04:

- Питающее напряжение 5 В,
- Рабочие параметры силы тока 15 мА,

- Сила тока в пассивном состоянии менее 2 мА,
- Обзорный угол 15°,
- Сенсорное разрешение 0,3 см,
- Измерительный угол 30°,
- Ширина импульса 10^{-6} с.

Датчик имеет четыре вывода:

- Питания +5В
- Trig – вывод сигнала входа
- Echo – вывод сигнала выхода
- GND – вывод «Земля»

«Принцип действия датчика (рисунок 13):

1) Для вывода датчика из режима ожидания, требуется подать стартовый импульс на вход Trig (положительный импульс длительностью 10 мкс);

2) Датчик генерирует 8 импульсов меандра с периодом 25 мкс (что соответствует частоте 40кГц) на ультразвуковой передатчик;

3) По спаду последнего сгенерированного импульса датчик устанавливает уровень логической «1» на выходе Echo, одновременно, датчик ждет получение отраженной ультразвуковой волны той же частоты на ультразвуковой приёмник;

4) После получения последнего импульса отраженной волны датчик переходит в режим ожидания, устанавливая уровень логического «0» на выходе Echo. Аналогичные действия будут совершены, если в течении 38 мс датчик не примет отраженную ультразвуковую волну. В результате время наличия логической «1» на выходе Echo равно времени прохождения ультразвуковой волны от датчика до препятствия и обратно.» [8]

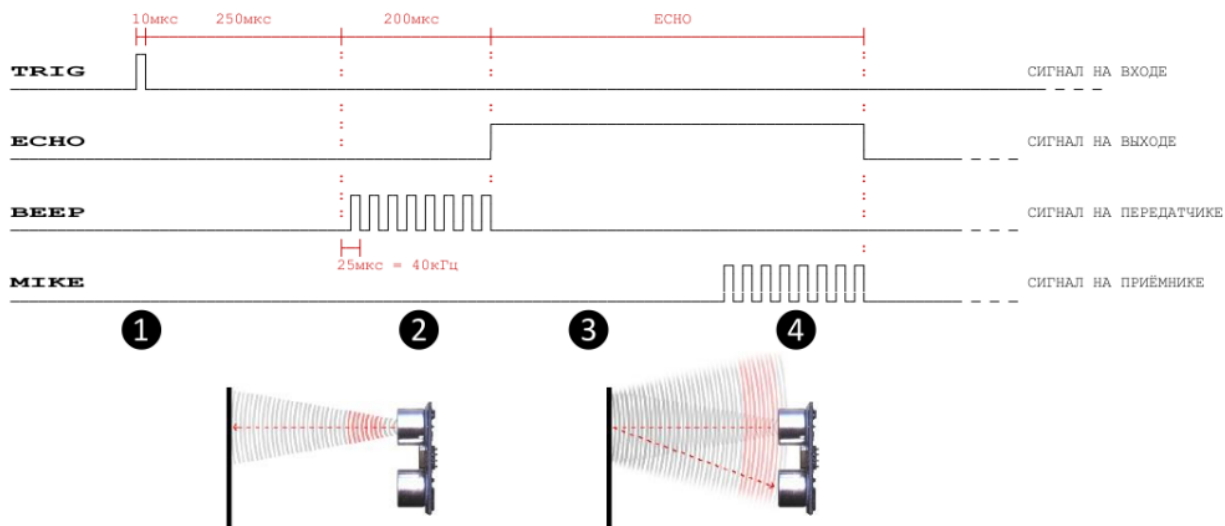


Рисунок 13 – Принцип действия

Направление импульсов в пространстве представлено на рисунке 14.

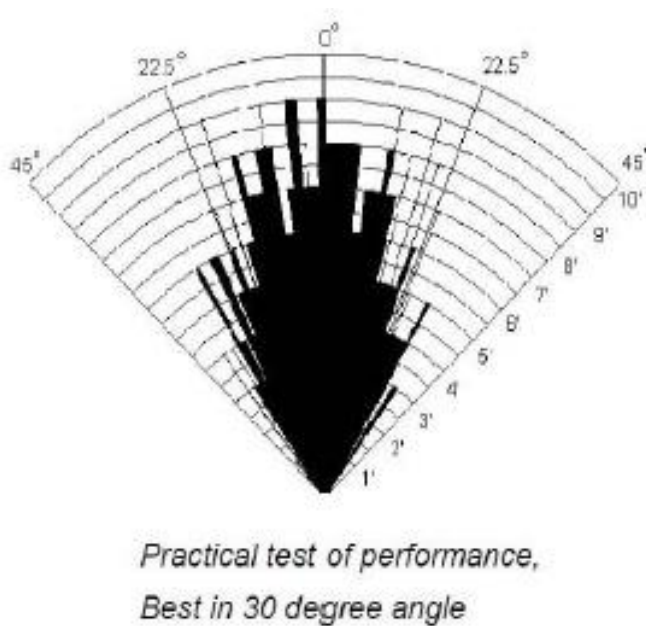


Рисунок 14 – Диаграмма направленности ультразвукового датчика

Рассмотрев оба датчика для нашего устройства, лучше подойдет ультразвуковой датчик HC-SR04. Выбор производился на основе технических характеристик. Основной характеристикой стала дистанция работы датчиков, для более приятной работы системы необходим запас по дальности, так как систему можно настроить программным способом под любую дистанцию.

2.3 Обоснования выбора светодиодной ленты

Важной частью является визуализация процесса работы системы. Для этого была выбрана адресная светодиодная лента WS2812B (рисунок 15).

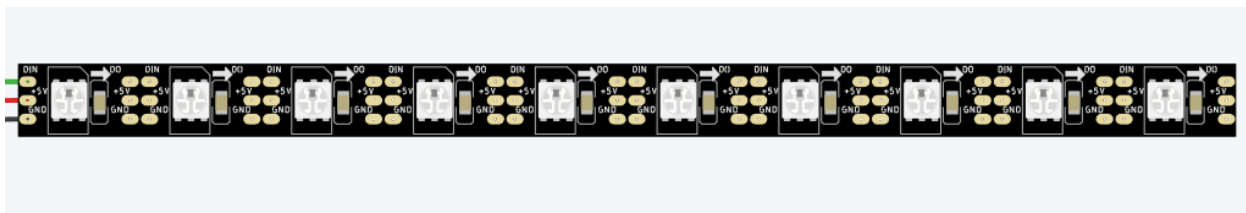


Рисунок 15 – Адресная светодиодная лента WS2812B

Характеристики ленты:

- Размер светодиода – 5*5 мм,
- Частота ШИМ – 400 Гц,
- Скорость передачи данных – 800 кГц,
- Размер данных – 24 бита на светодиод,
- Напряжение питания – 5 Вольт,
- Потребление при нулевой яркости – 1 мА на светодиод,
- Потребление при максимальной яркости – 60 мА на светодиод.

Оснащена лента адресными светодиодами (рисунок 16).

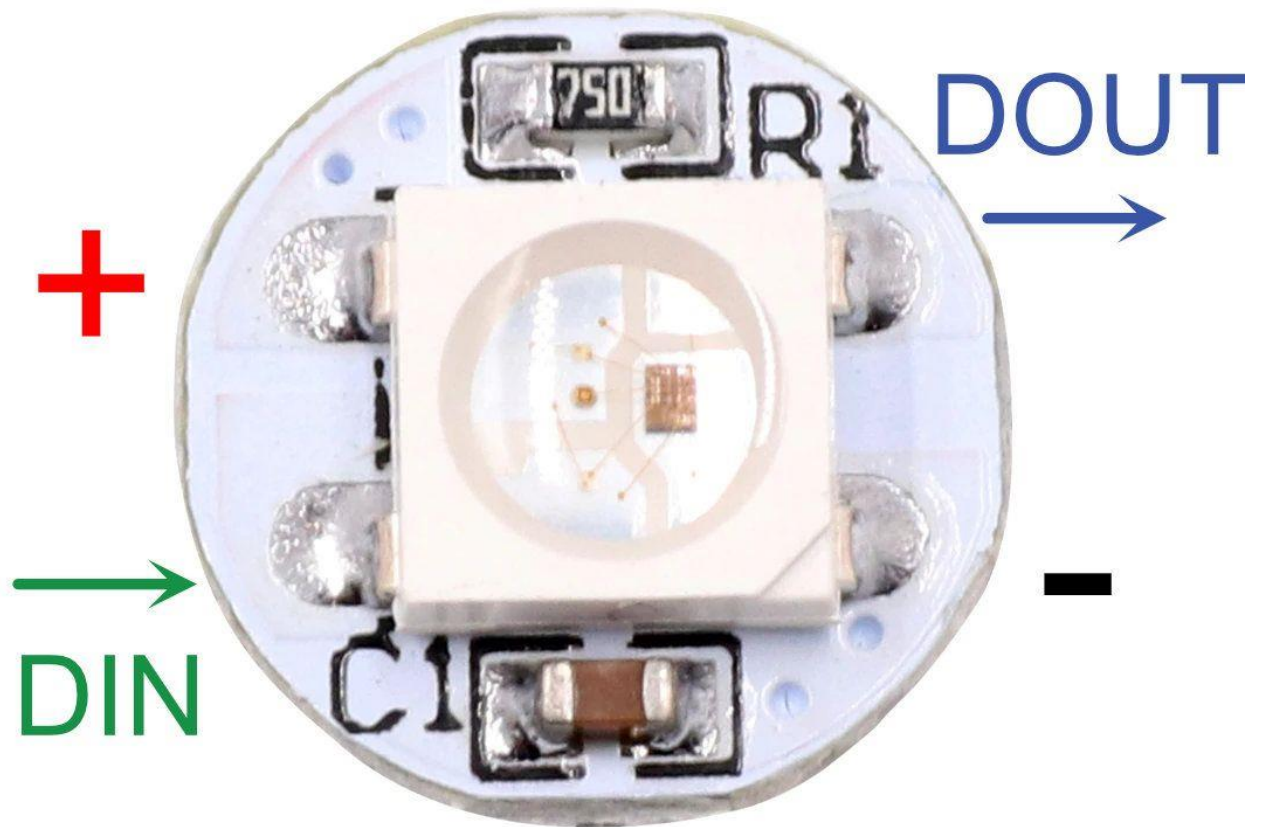


Рисунок 16 – Адресный светодиод

Светодиод оснащен двумя ножками питания, входом и выходом. Команда приходит на каждый цвет в виде последовательности 24 битов по 8 бит на цвет [9]. Данная последовательность приходит на вход светодиода. Чтобы светодиод понял последовательность бит, сигнал подается так, что ноль и единица отличаются по времени подачи: 0,4 мкс это ноль, а 0,85 мкс это единица. Время подачи для одного светодиода составляет 30 мкс. Пример передачи данных на один светодиод показан на рисунке 16.



Рисунок 17 – Передача данных

Данная лента является довольно популярной в разных проектах. Также отлично подходит под наши запросы, имея напряжения питания 5 Вольт и тока потребления на 10 светодиодов при максимальной яркости 0,6 Ампера. Поэтому решено было использовать именно эту светодиодную ленту.

2.4 Разработка структурной схемы

Структурная схема, представленная на рисунке 18, это совокупность элементарных объектов и связей между ними. Является одним из видов графической модели.

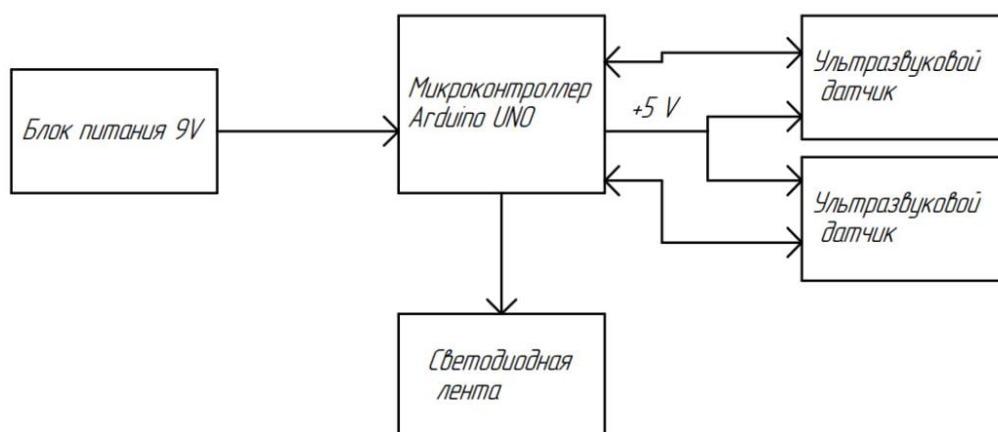


Рисунок 18 – Структурная схема

Как видно из рисунка, питание на микроконтроллер приходит с блока питания. Для стабильной работы системы на плату приходит 9 Вольт, а после нее идет питание 5 Вольт. К Arduino подключены два ультразвуковых датчика и светодиодная лента.

2.5 Разработка принципиальной схемы

В соответствии с ГОСТ 2.701–2008 принципиальная схема определяется как «схема, определяющая полный состав элементов и связей между ними и, как правило, дающая детальное представление о принципах работы изделия» [6]. Принципиальная схема представлена на рисунке 19.

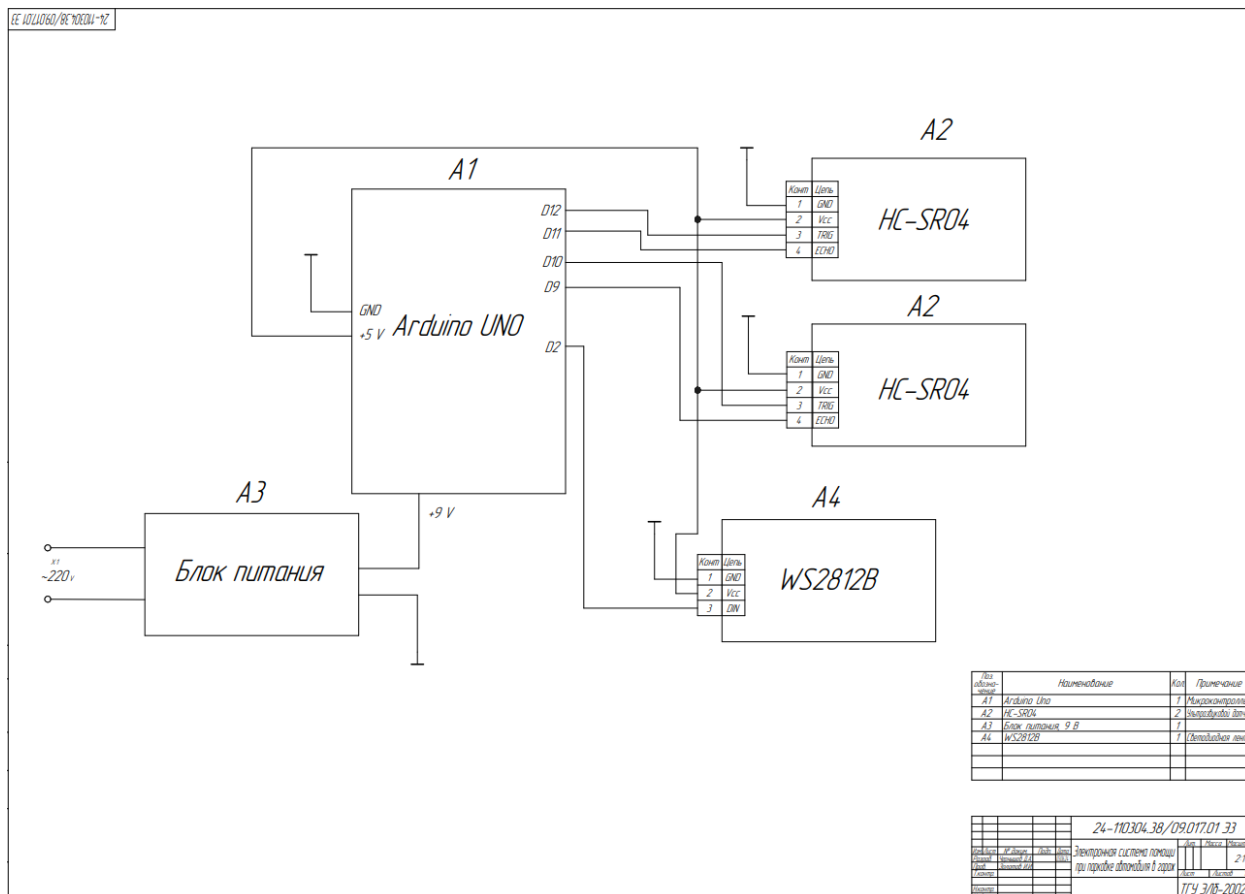


Рисунок 19 – Принципиальная схема

Данная схема показывает к каким выводам микроконтроллера будет подключена вся периферия и питание. Для увеличения надежности и гибкости используется два ультразвуковых датчика HC-SR04.

Выводы по разделу

В ходе анализа обзорных решений по выбору микроконтроллера были выявлены характерные достоинства, присущие выбранной аппаратной платформе:

– главным достоинством модели Arduino UNO является наличие разъема

POWERJACK, при помощи которого можно реализовать рекомендательное условие питающего напряжения +9 В;

В результате выбора датчика расстояния предложено использовать ультразвуковой датчик:

- датчик имеет низкую себестоимость;
- данный тип датчика расстояния обладает меньшей инерционностью по сравнению с другими датчиками расстояния, вследствие чего достигаются лучшие вибрационные характеристики;
- за счет продольного размещения преобразовательного элемента и проводящего листа, которые замыкаются на «землю», датчик имеет меньшую подверженность шумам.

В качестве итогового варианта по выбору светодиодной ленты предложена модель WS2812B по следующим составляющим:

- доступная цена и высокий рейтинг использования;
- низкое энергопотребление в плане управления с низковольтных устройств;

В соответствии с предложенной элементной базой разработаны структурная схема, отражающая систему управления помощи при парковке, а также схема электрическая принципиальная, иллюстрирующая все необходимые электрические соединения и электрические связи компонентов устройства.

3 Разработка алгоритма работы устройства

Перед разработкой управляющей программы устройства необходимо продумать алгоритм работы устройства. Каждое электронное устройство на микроконтроллере работает по определенному алгоритму, который проходит от начала до конца программного кода. По началу продумывается примерный алгоритм работы, который в результате приведет к нужной цели. В дальнейшем, основываясь на алгоритм, пишется код программы, который преобразует цепочку алгоритма в понятный для микроконтроллера язык.

На рисунке 20 представлена блок-схема алгоритма, которая представляет собой начало программы (1 частью) [19].



Рисунок 20 – Блок-схема алгоритма 1 часть

Данная часть алгоритма отвечает за начала работы системы, в которой происходит объявление переменных, инициализация ультразвуковых датчиков и светодиодной ленты. После чего система начинает измерять

дистанцию до препятствия и переводить ее в сантиметры. Это необходимо для упрощенного понимания дистанции. За счет этого и появляется возможность изменить переменные, отвечающие за остановку и предупреждения о скорой остановке.

При разработке алгоритма принимаются решения какие функции будут использованы в коде программы.

На рисунке 21 представлена блок-схема алгоритма, которая является серединой алгоритма (2 частью).

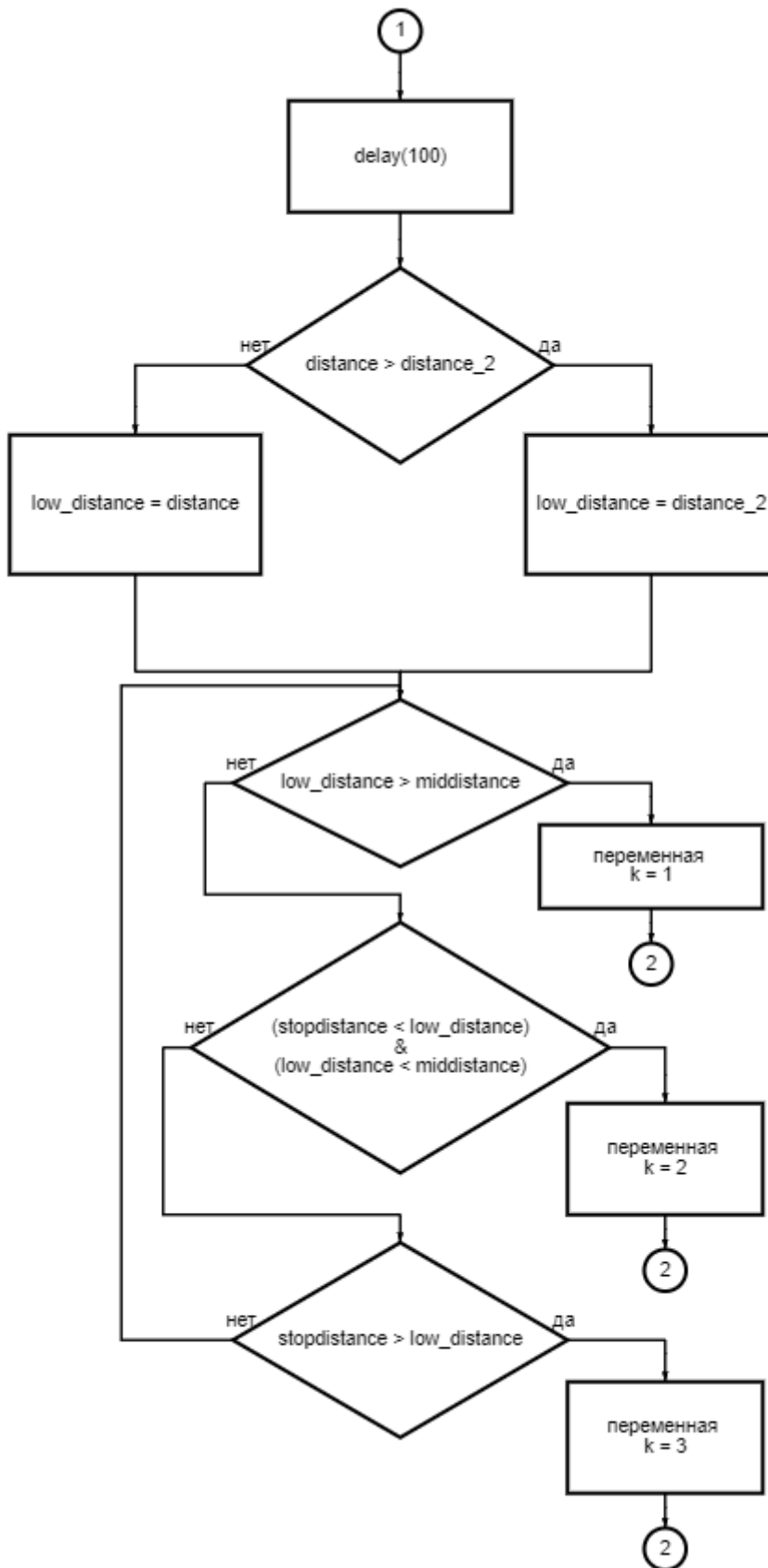


Рисунок 21 – Блок схема алгоритма 2 часть

Средняя часть алгоритма предназначена для определения наименьшего

значения с ультразвуковых датчиков, которое будет записываться в переменную. В дальнейшем эта переменная сравнивается с константами по дистанции, которые задаются в начале в зависимости от требования пользователя. В результате сравнения на выходе переменной k присваивается значение 1, 2 или 3. В дальнейшем алгоритме данная переменная будет определять работу светодиодной ленты. На рисунке 22 представлена блок-схема алгоритма, которая является заключительной частью (3 частью).

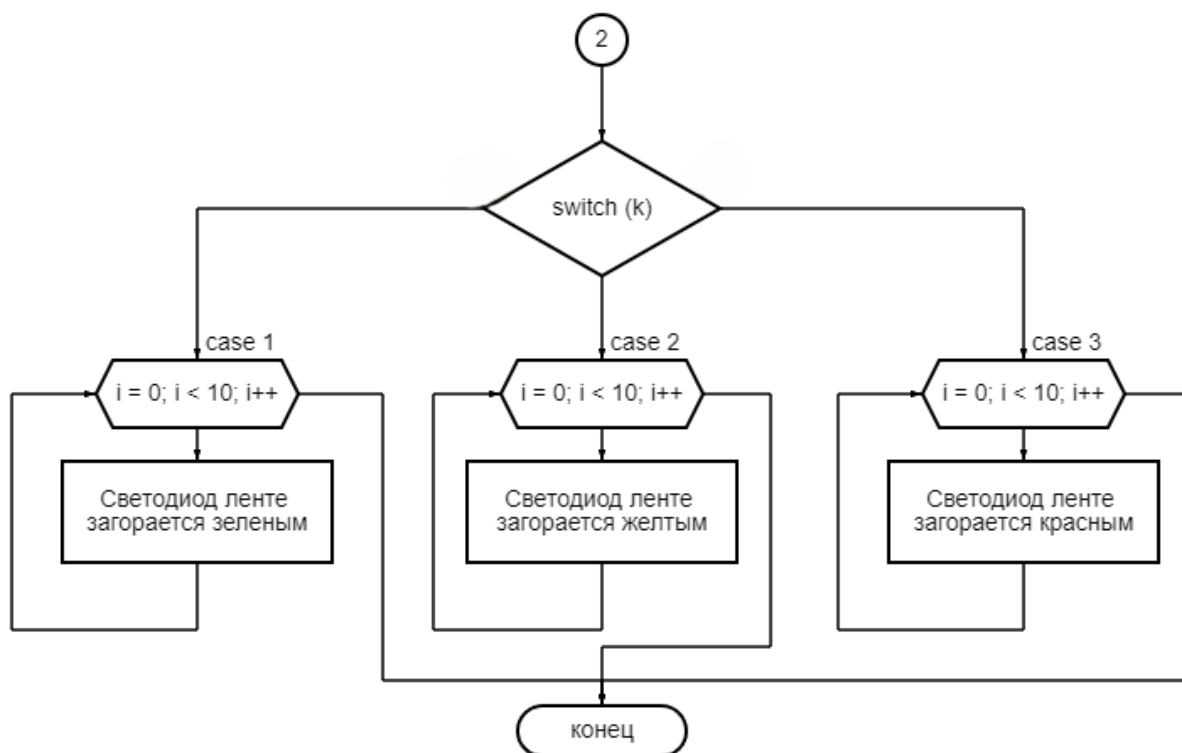


Рисунок 22 – Блок-схема алгоритма 3 часть

Как ранее сказало, переменная k определяет дальнейший действия алгоритма. Для этого можно использовать функцию switch () это функция производит сравнение значения переменной со своими значениями, которые указываются при написании кода программы. В результате выбора ключа k переменной происходит включение светодиодной ленты необходимым цветом.

Вывод по разделу

Результатом данного раздела является пошаговая проработка блок-схемы алгоритма управления устройством, состоящая из трех частей, каждая из которых отвечает за конкретную стадию программы. В результате проработки были продуманы функции, которые используются в программировании, и логика работы кода программы.

4 Разработка управляющей программы устройства

Для разработки программного обеспечения была использована программа Arduino IDE, которая работает на языке C ++ [16]. Сама программа упрощает работу на этом языке для того, чтобы неопытным пользователям было проще работать в этой среде. Но и также программа позволяет оформить код в стиле Си. Arduino IDE автоматически подключает необходимые библиотеки для правильной работы с микроконтроллером [14].

4.1 Подключение библиотек и объявление переменных

Для упрощения взаимодействия с компонентами Arduino существует множество библиотек, в которых хранятся сокращенные команды. Все эти библиотеки делаются другими пользователями и поддерживаются в актуальном состоянии.

Для нашего устройства необходима будет одна библиотека.

```
#include <Adafruit_NeoPixel.h>
```

«Библиотека предназначена для работы с различными светодиодными лентами и неопиксельными кольцами (NeoPixel Ring), написана на C++, поддерживает светодиодные ленты и неопиксельные кольца, основанные на светодиодах WS2812 и WS2811. Библиотека NeoPixel резервирует 3 байта оперативной памяти на каждый светодиод. По умолчанию библиотека настраивается на светодиоды WS2812, флаг NEO_GRB + NEO_KHZ800 в этом случае можно опустить. Если вы подключаете светодиодную ленту к порту 6, то номер порта можно не указывать. Таким образом обязательным параметром является только количество светодиодов в ленте.» [3]

Благодаря этой библиотеки мы получаем упрощенный вариант работы со светодиодной лентой. А именно позволяет подключить светодиодную ленту 3 командами.

```
#define LED_COUNT 10

#define LED_PIN 2

...

Adafruit_NeoPixel strip (LED_COUNT,LED_PIN,NEO_GRB +
NEO_KHZ800);
```

Вначале мы объявляем количество светодиодов на ленте и выход, к которому подключена светодиодная лента. Далее происходит объявление светодиодной ленты задается ее имя в скобках указывается константы, а именно количество светодиодов и выход, тип светодиодной ленты и ее модель. В результате мы получаем имя светодиодной ленты, по которому мы и будем с ней взаимодействовать.

Также для работы устройства необходимо иметь переменные, в которых будет храниться пороговое значение дистанции.

```
int stopdistance = 100;
```

```
int middistance = 160;
```

Первая переменная хранит в себе значение, при котором необходимо остановить машину. Вторая хранит значение служащие для предупреждения о приближение к месту остановки.

```
#define PIN_TRIG 12
```

```
#define PIN_ECHO 11
```

```
#define PIN_TRIG_2 10
```

```
#define PIN_ECHO_2 9
```

Данные константы хранят в себе информацию о подключение ультразвуковых датчиков к микроконтроллеру.

4.2 Программирование ультразвуковых датчиков

После объявления пинов для подключения датчиков их необходимо инициализировать. А именно указать какой пин будет использоваться для выхода сигнала, а какой для входа отраженного сигнала.

```
pinMode (PIN_TRIG, OUTPUT);
```

```
pinMode (PIN_ECHO, INPUT);
```

```
pinMode (PIN_TRIG_2, OUTPUT);
```

```
pinMode (PIN_ECHO_2, INPUT);
```

PIN_TRIG это пин для выхода сигнала, а PIN_ECHO для входа [17, 18]. Аналогично для второго датчика.

Так как принцип действия датчиков основан на отправке сигнала и ожидание его возврата. Для этого программным способом отправить сигнал.

```
digitalWrite (PIN_TRIG, LOW);
```

```
delayMicroseconds (5);
```

```
digitalWrite (PIN_TRIG, HIGH);
```

```
delayMicroseconds (10);
```

```
digitalWrite (PIN_TRIG, LOW);
```

```
duration = pulseIn (PIN_ECHO, HIGH);
```

Данная часть программы отвечает за отправку сигнала и запоминание через сколько времени сигнал вернулся обратно. Это время необходимо перевести в дистанцию.

```
distance = (duration / 2) / 29.1
```

Именно эта строчка программы отвечает за пересчет времени сигнала в сантиметры.

4.3 Программирование логики и светодиодной ленты

После подключения библиотеки для ленты и обозначения ее входов для подключения необходимо запустить ленту в работу. Для этого в коде программы, а именно в функции `void setup` прописываются следующие команды [1].

```
strip.begin();
```

Так как лента имеет короткое название, благодаря библиотеки, `strip` через это имея мы и работаем с ней. Данная команда отвечает за запуск светодиодной ленты.

```
strip.setPixelColor(i, strip.Color(255, 255, 0));
```

```
strip.show();
```

В дальнейшем именно эти две команды будут отвечать за подсвечивание светодиодов нужным цветом. В скобках первой команды первым указывается номер светодиода на ленте далее указывается цвет, которым светодиод должен загореться.

Для реализации работы системы необходимо было продумать как оно будет работать. Было решено для повышения надежности системы использовать два ультразвуковых датчика, но возникает вопрос на какой из них будет опираться микроконтроллер для работы.

```
if (distance > distance_2){  
    low_distance = distance_2;  
}  
  
else{  
    low_distance = distance;  
}
```

В этой части программы происходит выбор с какого датчика будет взято

значение дистанции до объекта. Основа выбора происходит на том с какого датчика выходит меньшее значение дистанции.

```
If (low_distance > middistance){  
    k = 1;  
}  
  
If ((stopdistance < low_distance)&(low_distance < middistance)){  
    k = 2;  
}  
  
if (stopdistance > low_distance){  
    k = 3;  
}
```

После производится сравнение считываемой дистанции и константных значений. В дальнейшем переменной присваивается значение на основе сравнений. Это необходимо для работы программы с оператором switch [2, 18].

```
switch (k){  
    case 1:  
        for(i = 0; i < 10; i++){  
            strip.setPixelColor(i, strip.Color(0, 255, 0));  
            strip.show();  
        }  
        break;  
    case 2:  
        for(i = 0; i < 10; i++){  
            strip.setPixelColor(i, strip.Color(255, 255, 0));
```

```

        strip.show();
    }
    break;
case 3:
    for(i = 0; i < 10; i++){
        strip.setPixelColor(i, strip.Color(255, 0, 0));
        strip.show();
    }
    break;
}
}

```

Именно в этой части и реализуется вся система, опираясь на значение переменной *k*, происходит выбор ключа. В каждом ключе прописан порядок действия. Так как светодиодная лента адресная, появляется необходимость обращаться к каждому светодиоду и включать его определенным цветом. За это отвечает цикл `for`.

```

for(i = 0; i < 10; i++){
    strip.setPixelColor(i, strip.Color(0, 255, 0));
    strip.show();
}

```

Каждый светодиод на ленте имеет свой номер начиная с 0. Именно для этого цикл `for` и необходим. На нашей ленте таких светодиодов 10 штук, как и повторений в цикле это необходимо чтобы все светодиоды загорелись необходимым цветом.

Вывод по разделу

В этой главе с использованием программы Arduino IDE была разработана управляющая программа на основе используемых периферийных устройств. Описана подключенная библиотека для светодиодной ленты.

Также в главе пояснены все используемые функции и переменные используемые в программе. Разобрана логика работы программы и подключения ультразвуковых датчиков и светодиодной ленты.

После разработки кода он был загружен в микроконтроллер для проверки его работоспособности. Результатом работы является успешная работа системы.

5 Проектирование модели устройства и отладка в пакете автоматизированного проектирования

Проектирование модели — это особо важный этап для реализации какого-либо устройства или системы. Именно модель поможет понять, как себя может повести устройство, а также выявит неисправности и ошибки при моделировании.

Для моделирования работы аппаратной платформы Arduino, был выбран сайт Tinkercad, представляющий собой пакет автоматизированного проектирования. В нем, для целей моделирования, реализовано на уровне моделей большинство периферийных устройств аппаратной платформы Arduino. Сайт позволяет создавать проекты практически любой сложности.

Модель устройства представлена на рисунке 23.

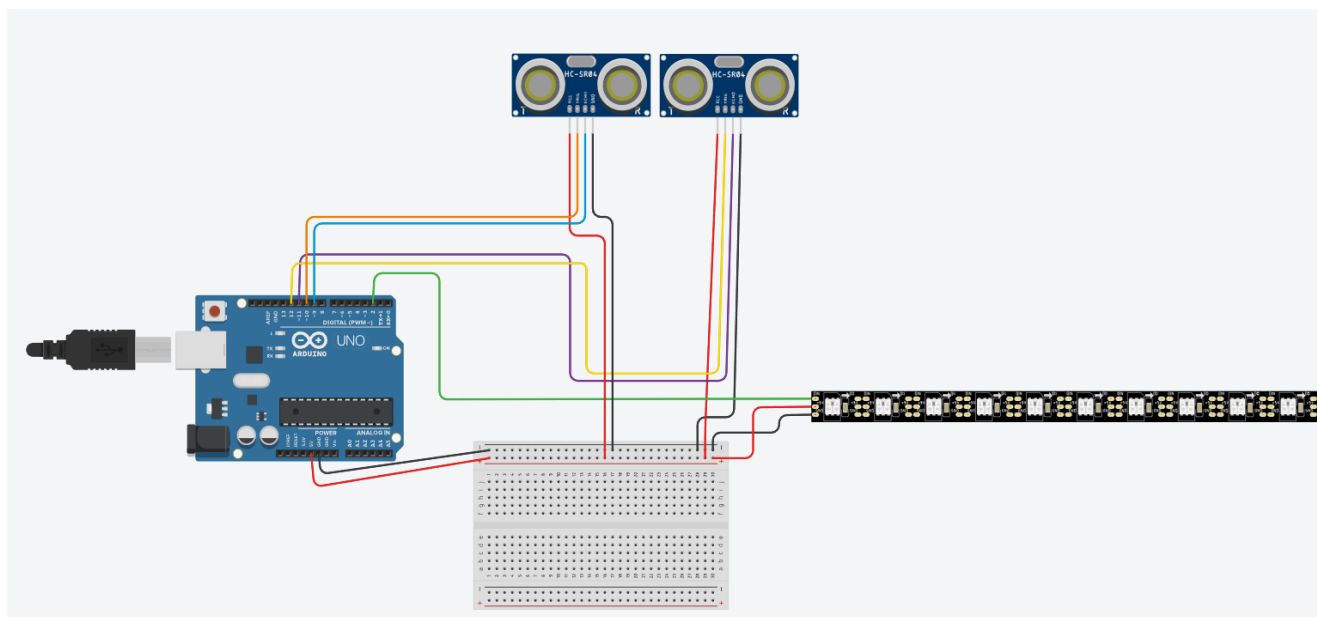


Рисунок 23 – Модель устройства

Благодаря возможности промоделировать устройство прямо на сайте можно исправлять код программы и сразу его проверять. А также можно увидеть результат работы устройства.

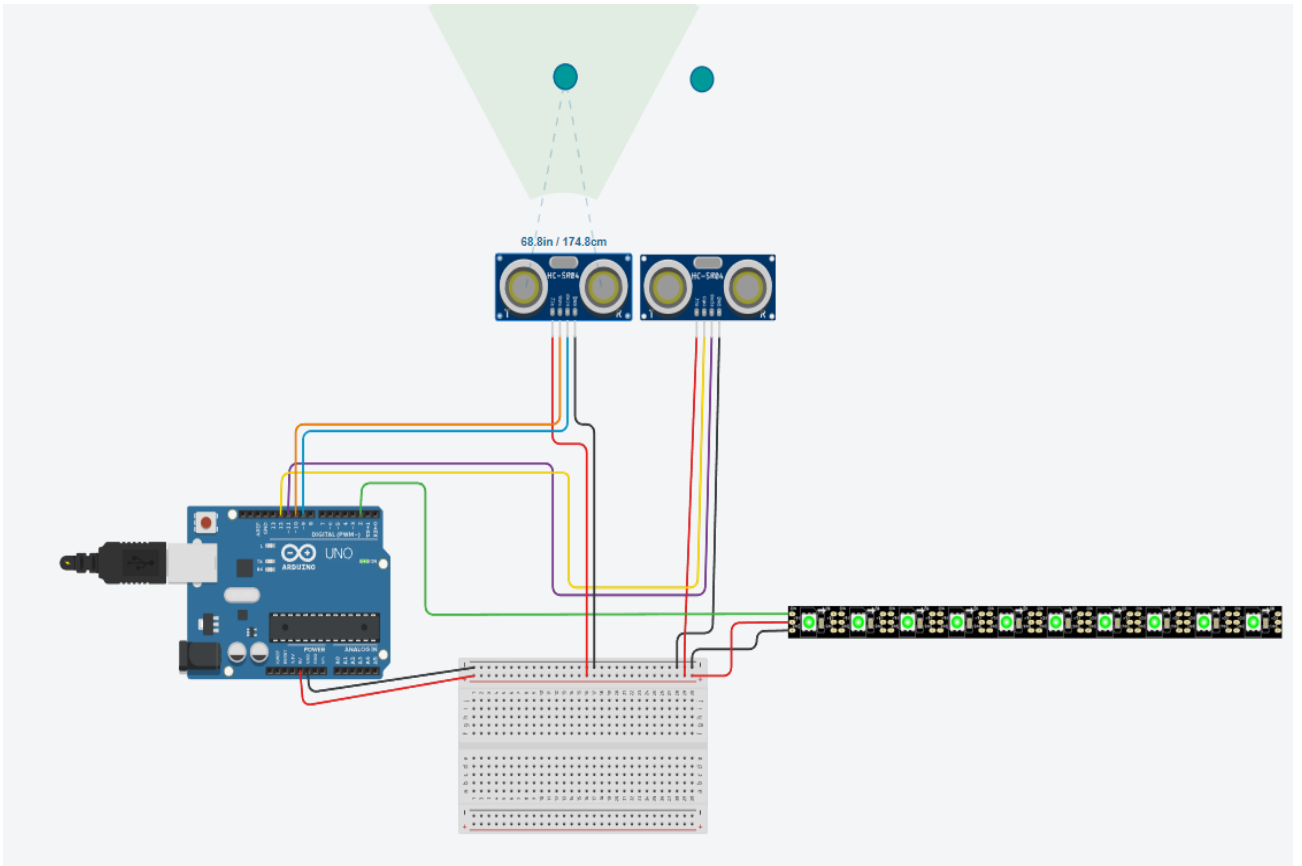


Рисунок 24 – Результат работы 1 ключа программы

Если дистанция до объекта с двух датчиков более 160 сантиметров, то в результате мы получаем зеленый цвет светодиодов.

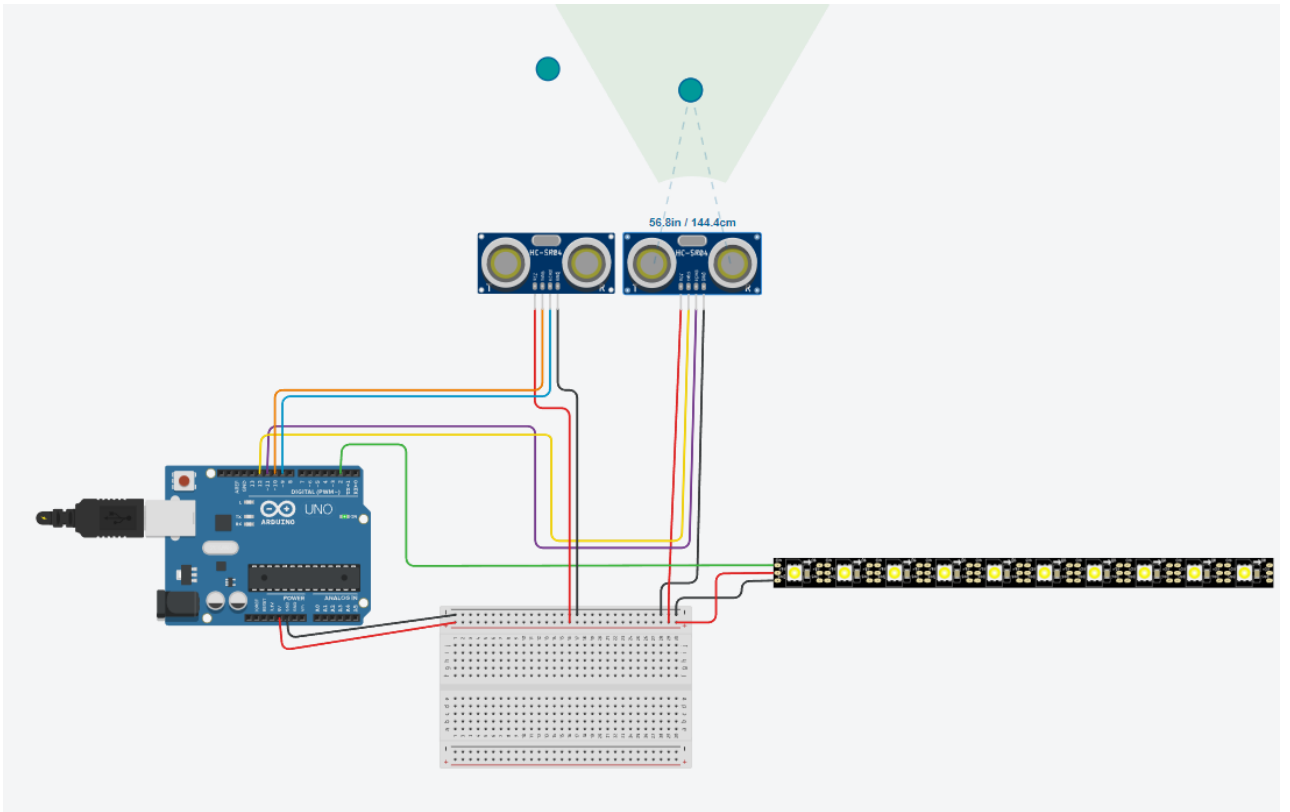


Рисунок 25 – Результат работы 2 ключа программы

В данном случае, как только один датчик начнет показывать дистанцию менее 160 сантиметров, цвет светодиодов переключится на желтый, что будет означать о скором приближение к месту остановки.

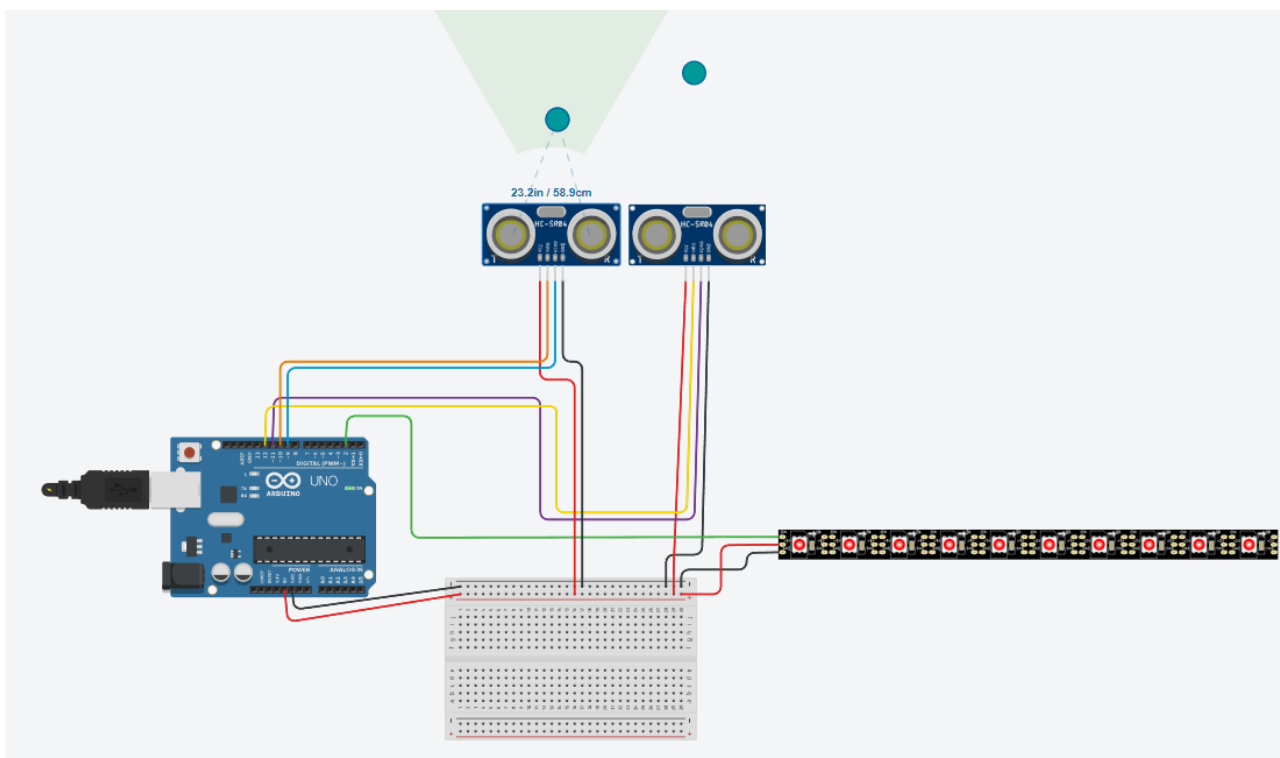


Рисунок 26 – Результат работы 3 ключа программы

Если объект приблизится менее чем на 100 сантиметров мы получим красный предупреждающий сигнал об остановке.

Код программы позволяет без особых трудностей изменять дистанции срабатывания, которые отвечают за изменения цвета светодиодной ленты по мере приближения к препятствию. Реализовано это через две переменные, значение которых как раз и отвечают за изменения цвета светодиодов. Это позволяет не трогать полностью код программы, а лишь изменить значения переменных на необходимые и индивидуальные.

Возможности, которые реализованы на сайте Tinkercad, позволяют заменять периферийные устройства из доступных. Для данной системы на сайте можно увеличить количество светодиодов, используемых лентой.

Выводы по разделу

В данном разделе была спроектирована и отлажена математическая модель в программной среде Tinkercad. Проведена симуляция работы ее

алгоритма с разной дистанцией до препятствия и, соответственно, в зависимости от значения дистанции применяется определенный ключ программы.

Также в программе заложена возможность изменения дистанции остановки и предупреждающей дистанции.

Заключение

В ходе настоящей выпускной квалификационной работы была разработана электронная система помощи при парковке автомобиля в гараж. Система оснащена двумя ультразвуковыми датчиками и адресной светодиодной лентой для визуализации приближения к препятствию.

Для данной системы были разработаны все необходимое электронные и электрический связи, были выбраны наилучшие компоненты, которые отлично подходят под условия эксплуатации. При этом устройство получило высокую надежность, благодаря проверенной периферии, а также имеет большую гибкость, как по настройки дистанции, так и по запросам расположения датчиков от пользователя.

Перед программированием системы был разработан алгоритм работы устройства, благодаря которому система получила логику работы. Основываясь на ней, были созданы блок-схемы с последовательностью действий устройства.

В дальнейшем была написана управляющая программа, основанная на алгоритме работы. В разделе с программированием полностью раскрыта суть кода программы, благодаря чему понимание работы программы упрощается в разы. В результате получили хорошо адаптивную систему, использующую две переменных, которые позволят быстро настроить нужные для пользователя дистанции парковки.

После полной разработки схемотехнических решений и программной части, мы смоделировали модель используя сайт Tinkercad, представляющий собой пакет автоматизированного проектирования. Смоделировав модель устройства, мы получили абсолютно работающее по назначению устройство.

Список используемой литературы

1. Белов, А.В. «Микроконтроллеры AVR: от азов программирования до создания практических устройств» / А. В. Белов. - СПб.: Наука и техника, 2016. - 544 с.
2. Бьерн Страуструп. «Язык программирования C++»: - 369 с.
3. Библиотека для адресной светодиодной ленты WS2812B [Электронный ресурс] // URL: https://github.com/adafruit/Adafruit_NeoPixel (дата обращения 08.05.2024)
4. Водовозов А.М. «Микроконтроллеры для систем автоматики»: Учебное пособие / А.М. Водовозов. - Вологда: ВоГТУ, 2002. - 123 с.
5. Гололобов В. Н. «ARDUINO. Просто о сложном»: - 368 с.
6. ГОСТ 2.701—2008 «Единая система конструкторской документации. Схемы. Виды и типы. Общие требования к выполнению»: – Введён 01.07.2009 г.
7. Петин В. А., Биняковский А.А. «Практическая энциклопедия Arduino»: - 2-е изд. - ДМК-Пресс, 2020. - 166 с.
8. Документация на ультразвуковой датчик HC-SR04 [Электронный ресурс] // URL: <https://iarduino.ru/lib/8c65ac1b85b79ef3fed8c9a9fa699147.pdf> (дата обращения 01.05.2024)
9. Документация на адресную светодиодную ленту WS2812B [Электронный ресурс] // URL: <https://amperkot.ru/static/3236/uploads/datasheets/WS2812B-LED-datasheet.pdf> (дата обращения 07.05.2024)
10. Документация на регулятор напряжения SPX1117M3-L-5 [Электронный ресурс] // URL: <https://www.farnell.com/datasheets/1468195.pdf> (дата обращения 05.05.2024)
11. Документация на Arduino UNO [Электронный ресурс] // URL: <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf> (дата

- обращения 28.04.2024)
12. Документация на Arduino Nano [Электронный ресурс] // URL: <https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf> (дата обращения 29.04.2024)
 13. Документация на инфракрасный датчик GP2Y0A021YK0F // URL: https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf
 14. Улли Соммер «Программирование Микроконтроллерных Плат Arduino/Freeduino»: - СПб.: БХВ-Петербург, 2012. - 256 с.
 15. Arduino [Электронный ресурс] // URL: <https://www.arduino.cc/> (дата обращения 28.04.2024)
 16. «Arduino IDE» [Электронный ресурс]. «Download the Arduino IDE» URL: <https://www.arduino.cc/en/main/software> (дата обращения 30.04.2024).
 17. Simon Monk «Programming Arduino: Getting Started with Sketches»: 2nd edition. 2016. 191p. ISBN 9781259641633
 18. James A. Langbridge. «Arduino Sketches. Tools and Techniques for Programming Wizardry», 2015. 480p. ISBN 9781118919606
 19. Explain Algorithm and Flowchart with Examples [Электронный ресурс] // URL: <https://www.edrawsoft.com/explain-algorithm-flowchart.html> (дата обращения 06.06.2024)
 20. Electronic Components Datasheets [Электронный ресурс] // URL: <http://www.datasheets.ru> (дата обращения 28.04.2024)

Приложение А

Программный код алгоритма работы

```
#include <Adafruit_NeoPixel.h>

#define PIN_TRIG 12

#define PIN_ECHO 11

#define PIN_TRIG_2 10

#define PIN_ECHO_2 9

#define LED_COUNT 10

#define LED_PIN 2

int duration, distance, duration_2, distance_2;

Adafruit_NeoPixel strip(LED_COUNT,LED_PIN,NEO_GRB + NEO_KHZ800);

int stopdistance = 100;

int middistance = 160;

int i, k, low_distance;

void setup() {

  pinMode(PIN_TRIG, OUTPUT);

  pinMode(PIN_ECHO, INPUT);

  pinMode(PIN_TRIG_2, OUTPUT);

  pinMode(PIN_ECHO_2, INPUT);

  strip.begin();

}
```


Продолжение приложения А

```
void loop(){  
    digitalWrite(PIN_TRIG, LOW);  
    delayMicroseconds(5);  
    digitalWrite(PIN_TRIG, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(PIN_TRIG, LOW);  
    duration = pulseIn(PIN_ECHO, HIGH);  
    distance = (duration / 2) / 29.1;  
    delay(10);  
    digitalWrite(PIN_TRIG_2, LOW);  
    delayMicroseconds(5);  
    digitalWrite(PIN_TRIG_2, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(PIN_TRIG_2, LOW);  
    duration_2 = pulseIn(PIN_ECHO_2, HIGH);  
    distance_2 = (duration_2 / 2) / 29.1;  
    delay(100);  
    if (distance > distance_2){  
        low_distance = distance_2;  
    }  
    else{  
        low_distance = distance;  
    }  
}
```

Продолжение приложения А

```
if (low_distance > middistance){  
    k = 1;  
}  
  
if ((stopdistance < low_distance)&(low_distance < middistance)){  
    k = 2;  
}  
  
if (stopdistance > low_distance){  
    k = 3;  
}  
  
switch (k){  
case 1:  
    for(i = 0; i < 10; i++){  
        strip.setPixelColor(i, strip.Color(0, 255, 0));  
        strip.show();  
    }  
break;  
case 2:  
    for(i = 0; i < 10; i++){  
        strip.setPixelColor(i, strip.Color(255, 255, 0));  
        strip.show();  
    }  
break;
```

Продолжение приложения А

case 3:

```
for(i = 0; i < 10; i++){  
    strip.setPixelColor(i, strip.Color(255, 0, 0));  
    strip.show();  
}  
break;  
}  
}
```