

## Содержание

Введение.....	4
1. Техническое задание.....	6
1.1 Назначение разработки и область применения.....	6
1.2 Технические требования.....	7
2. Анализ технического задания.....	8
2.1 Выбор варианта реализации.....	8
2.2 Выбор операционной системы для персонального компьютера.....	10
2.3 Выбор языка программирования.....	10
2.4 Выбор среды разработки.....	10
3. Разработка системы на структурном уровне.....	11
3.1 Структура робототехнического устройства.....	11
3.2 Выбор варианта реализации подсистем.....	12
3.3 Структурная схема устройства.....	13
3.4 Принцип работы основного датчика.....	14
3.5 Компоновка и подключение.....	18
4. Разработка алгоритмов.....	22
4.1 Разработка алгоритма управления оператором.....	22
4.2 Разработка алгоритма обхода лабиринта.....	24
5. Разработка системы на программном уровне.....	27
5.1 Разработка программных средств для управления оператором.....	27
5.2 Разработка программных средств для обхода лабиринта.....	29
6. Тестирование разрабатываемой системы.....	31
Заключение.....	32
Литература.....	33
Приложения.....	34

Подп. и дата	
Инв. № дубл.	
Взам. Инв. №	
Подп. и дата	
Инв. № подл.	

					ВКР-НГТУ-09.03.01-(14-ВМ)-003-				
					2019(ПЗ)				
Изм	Лист	№ докум.	Подп.	Дата	Программно-аппаратная система обнаружения препятствий для мобильного робота  Пояснительная записка	Лит.	Лист	Листо	
Разраб.	Корельский						3	50	
Пров.	Гай В.Е.								
Н. контр.									
Утв.	Митиньков								
						НГТУ им. Р.Е. Алексеева			

## Введение

Тенденция современного развития информационных технологий и робототехники ставят перед разработчиками всё новые и новые задачи. Одна из таких задач – определение препятствий при управлении мобильными роботами на расстоянии.

Одним из перспективных направлений робототехники являются роботы телеприсутствия. Основной задачей данных роботов является наблюдение за каким-либо событием, с возможностью повлиять на данное событие, взаимодействие с окружающими людьми по средствам передачи визуальных и звуковых данных. Подобные роботы уже имеют широкое распространение во многих сферах деятельности человека: медицина, военная, научно – исследовательская и т.д.

В современном обществе разработкой различного вида роботов можно заниматься в домашних условиях. Этому способствует широкое распространение информации в интернете, телевидении и прессе, лёгкая доступность элементной базы, для разработки роботов, и её многообразие.

Подобным проектом стала разработка робота телеприсутствия ElcBot. Данный робот может иметь широкое применение в гражданской сфере деятельности человека, например: для обеспечения возможности присутствия детей, не имеющих возможность физически посещать занятия в школе по каким-либо причинам, присутствовать на занятиях, общаться со сверстниками тем самым поддерживая коммуникацию с окружающими. Или использование данной разработки в пожарно – охранной сфере путём патрулирования робота в помещении, преимуществом перед стационарными приборами, выполняющими ту же функцию, в скорости и точности реагировании в случае возникновения нештатной ситуации, к примеру, на заводах или иных масштабных территориях, при удалённой нахождении охраны.

Для выполнения выше названных функций требуется система по определению препятствия, которая позволит эффективно управлять данным роботом или для автоматического движения. Подобные системы успешно используются во многих автономных роботизированных устройствах, но на данный момент она является неотъемлемой частью каждого из них. Несмотря на то, что суть системы заключается в распознавании препятствий, сами системы различаются по следующим критериям:

1. Специфики использования роботов изначально диктуют основополагающие условия для создания системы. Таковыми являются условия

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	Введение	
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)	Лист
						4

эксплуатации, изначальный бюджет разработки, количество специалистов, привлечённых к разработке и т.д.

2. Схожие по своему функционалу роботы различаются между собой начиная от банального количества датчиков, заканчивая языками программирования, использованные в процессе написания кода программы.

3. Даже если удастся найти подходящую по требованиям систему возникнет проблема по внедрению её в разрабатываемый проект, так как методы управления самим роботом будут отличаться от копируемой системы. В случае если система совпадёт требованиями проекта проектом возникнет проблема авторских прав по использованию.

На фоне выше озвученных проблем, разработка подобной системы с нуля для проекта выглядит менее проблематичной, чем поиск и подстраивание готовой системы.

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)					Лист
										5

## 1. Техническое задание

### 1.1 Назначение разработки и область применения

Разрабатываемая система предназначена для робота телеприсутствия ElcBot при движении в 2-х режимах:

Управление оператором с ЭВМ;

Обход лабиринта;

Движение обеспечивается программно – аппаратными модулями, расположенными на компьютере и непосредственно на роботе.

В любом из выбранных режимах основная задача системы не позволять роботу врезаться в препятствие. Дополнительной функциями являются: в режиме обхода лабиринта – следование вдоль стены. В дальнейшем могут быть добавлены иные функции.

Целью проекта робота телеприсутствия ElcBot - создание прототипа робота для отладки алгоритмов управления, обработки данных, компьютерного зрения и распознавания звуковых сигналов. Разрабатываемая система позволит расширить функциональные возможности остальных систем, входящих в состав проекта.

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)					Лист
										6

## 1.2 Технические требования

Так как разрабатываемая система предназначена для проекта робота телеприсутствия ElcBot, то выбор аппаратно – технических средств ограничивается совместимыми с текущей системой:

Аппаратное:

- Плата Arduino Mega;
- Одноплатный компьютер Raspberry Pi B + (8Гб);
- Мотор стеклоподъемника двери (VW Polo);
- WiFi модуль для Raspberry Pi B +;
- Питание - элементы 18650;
- Модель адаптера;
- Модель колеса;
- Датчика HC – SR04.

Программное обеспечение:

- Robot Operating System;
- Raspbian Jessie;
- Arduino IDE;
- Python 3.

Разрабатываемая система должна обладать следующими функциями:

- определение препятствия на расстоянии до 2 метров;
- передавать сигналы прерывания управления, для предотвращения столкновения с препятствием.

Изн. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	<div>ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)</div>					Лист
Изм.	Лист	№ докум.	Подп.	Дата						7

## 2.1 Выбор варианта реализации

1. Одноплатный компьютер Raspberry Pi: основным критерием выбора данной модели его цена. Так как все необходимы функции по обработки и передачи данных, в разрабатываемой системе, способны выполнять большинство подобных одноплатный компьютер. Анализ рынка показал, что данный компьютер, подходящий под ОС Windows, является наиболее выгодным выбором, так как Raspberry Pi средняя цена на рынке составляет 3 000 рублей, а ближайший аналог от 5 000 рублей.

3. Мотор стеклоподъемника двери (VW Polo): В виду наличия мотор стеклоподъемника и высокой стоимости альтернативных электромоторов на рынке, было решено использовать имеющиеся.

5. Датчика НС – SR04: Для определения расстояния на выбор были предложены 3 вида датчиков: инфракрасный, ультразвуковой и лазерный.

Инфракрасные датчики являются довольно точными на малых расстояниях, а техническим заданием предусмотрено распознавание препятствий на расстоянии до 2 метров, а по своей специфике работы имеют большое рассеивание луча с увеличением расстояния. Это приводит к низкому качеству определения дистанции до препятствия. На рынке имеются инфракрасные датчики способные с высокой точностью определять препятствия на расстоянии до 5 метров, но они значительно дороже своих аналогов.

Наилучшие, в своём классе, по определению расстояния являются лазерные датчики расстояния. Даже самые простые из них способны с высокой точностью определять расстояние до препятствия. Но основным недостатком данных датчиков является высокая стоимость относительно аналогов.

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	3. Мотор стеклоподъемника двери (VW Polo): В виду наличия мотор
					стеклоподъемника и высокой стоимости альтернативных электромоторов на рынке, было
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	решено использовать имеющиеся.
					4. Драйвер мотора на L298N: Для возможности управлять двигателями постоянного
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	тока, регулировать скорость и направление вращения колёс. Драйвер мотора на L298N
					является одним из самых простых и дешёвых решений поставленной задачи.
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	5. Датчика НС – SR04: Для определения расстояния на выбор были предложены 3
					вида датчиков: инфракрасный, ультразвуковой и лазерный.
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	Инфракрасные датчики являются довольно точными на малых расстояниях, а
					техническим заданием предусмотрено распознавание препятствий на расстоянии до 2 метров,
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	а по своей специфике работы имеют большое рассеивание луча с увеличением расстояния.
					Это приводит к низкому качеству определения дистанции до препятствия. На рынке
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	имеются инфракрасные датчики способные с высокой точностью определять препятствия
					на расстоянии до 5 метров, но они значительно дороже своих аналогов.
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	Наилучшие, в своём классе, по определению расстояния являются лазерные датчики
					расстояния. Даже самые простые из них способны с высокой точностью определять
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	расстояние до препятствия. Но основным недостатком данных датчиком является высокая
					стоимость относительно аналогов.
					ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)
					Лист
					8

В виду отсутствия альтернативы был выбран тип ультразвукового датчика. Самым распространённым, дешёвым и подходящим к техническому заданию является датчик HC – SR04. Данный датчик также имеет ряд технических преимуществ перед инфракрасными и лазерными датчиками. Так как основной принцип работы ультразвукового датчика — это эхо-локация (сонар), основанной на природных возможностях животных, таких как летучие мыши или дельфины, распространению звуковых волн, в отличие от световых волн. колебания показателей влажности и температуры не сильно оказывает влияние на точность определения препятствий. В отличие от лазерного дальномера способен определять препятствия при ярком освещении и поверхности, пропускающие свет, такие как стеклянные двери.

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)					Лист
										9

## 2.2 Выбор операционной системы для персонального компьютера

Выбранной операционной системой для разработки системы выбрана Windows по следующим показателям:

1. Данная ОС выбрана основной для разработки робота телеприсутствия ElcBot;
2. Windows является наиболее распространённой ОС в мире;
3. Удобство и широкий выбор инструментов для разработки и программирования различных систем.

## 2.3 Выбор языка программирования

Выбор языка программирования ограничивается выбранными аппаратными средствами, используемыми в проекте робота телеприсутствия ElcBot и для реализации системы по определению препятствий. Основной платой в разрабатываемом проекте является Arduino Mega и для программирования используется язык, основанный C/C++, но существуют небольшие отличия от изначального варианта в виде встроенной библиотеки AVR Libc.

## 2.4 Выбор среды разработки

Средой разработки системы будет компилятор Arduino IDE состоящий из встроенного текстового редактора программного кода, области сообщений, окна вывода текста (консоли), панели инструментов с кнопками часто используемых команд и нескольких меню. Для загрузки программ и связи среда разработки подключается к аппаратной части Arduino. Написанная программа называется «скетч». Arduino IDE работает во всех распространённых операционных системах.

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)					Лист
										10



### 3. Разработка системы на структурном уровне

#### 3.1 Структура разрабатываемой системы

Разрабатываемая система предназначена для робота телеприсутствия ElcBot, и в общей структуре робота она будет относиться к подсистеме управления, но фактически будет подключена к исполнительной подсистеме. Взаимодействие разрабатываемой системы с пользовательским интерфейсом будет возможно только через исполнительную систему. Для взаимодействия

На основе анализа системы взаимодействия между основными подсистемами можно представить структуру в виде схемы, представленной на рис.1.

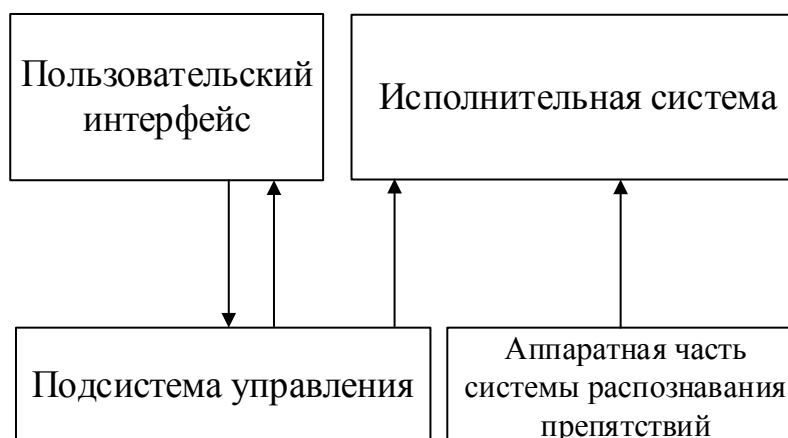


Рисунок 1. Структурная схема разрабатываемой системы.

Рисунок 1. Структурная схема разрабатываемой системы.						
Инв. № подл.	Подп. и дата				Лист	
	Инв. № дубл.					
	Взам. Инв. №					
	Подп. и дата					
					ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)	11
Изм.	Лист	№ докум.	Подп.	Дата		

### 3.2 Выбор варианта реализации подсистем

Разберём структурную схему более подробно. К пользовательскому интерфейсу относится компьютер с предустановленным программным обеспечением для передачи сигналов на подсистему управления, посредством подсистемы связи, реализованной в виде беспроводной связи WiFi.

Подсистема связи с одноплатным компьютером Raspberry Pi позволяет использовать протокол SSN. Этот протокол в дальнейшем позволит использовать более простое устройство, с предустановленным программным обеспечением, для управления роботом. Например, подобным устройством может выступать смартфон или планшет.

Исполнительная система, к которой относится каркас, электроприводы для колёс, колёса, драйвер мотора, соединена с подсистемой управления и аппаратной частью системы распознавания препятствий, представленной в виде системы из 9 датчиков HC – SR04, расположенных по периметру робота, по средствам подсистемы связи, представленной в виде медных проводов.

Плата Arduino Mega выступает связующим звеном между подсистемами. Её можно отнести как управляющей подсистеме, так и к исполнительной. Mega является основной платой выполняющей основной код управления роботом, по средствам поступления управляющих сигналов от Raspberry Pi и датчиков HC – SR04. На основании полученных управляющих сигналов формирует сигналы, для управления моторами.

В дальнейшем все подсистемы могут быть заменены на более производительные и с расширенным функционалом. К примеру, исполнительную подсистему можно будет на связать по средствам беспроводной связи с пользовательским интерфейсом для передачи данных о состоянии различных частей исполнительной системы, минуя управляющую систему. Это позволит не занимать рабочие циклы управляющей подсистемы передачей данных, тем самым сохраняя качество соединения и пересылки управляющих сигналов.

В частности, для разрабатываемой системы это позволило бы организовать совместно с системой компьютерного зрения графический интерфейс, на который отображал положение робота относительно препятствий и выводить предупреждение о столкновении.

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)					Лист
										12

### 3.3 Структурная схема устройства

Согласно описанию подсистем робота телеприсутствия ElcBot, структурную схему можно представить следующим образом:



Рисунок 2. Структурная схема устройства.

При дальнейшей модернизации структурная схема может претерпеть значительные изменения не только по способу взаимодействия, но и по составу подсистем. Одним из первых таковых изменений будет обратная связь между Arduino Mega и одноплатный компьютер Raspberry Pi для тестирования возможности передачи данных на компьютер, не используя дополнительных плат расширения. Одной из таких является Arduino. Arduino wi-fi shield - плата расширения на базе ESP8266 (Esp-12E)

Изн. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	<div>ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)</div>					Лист
										13
Изн.	Лист	№ докум.	Подп.	Дата						



Рисунок 3. Внешний вид ультразвуковых датчиков.

Датчик оснащен четырьмя выводами (стандарт 2,54 мм):

- Контакт питания положительного типа – +5В;
- Trig (T) – выход сигнала входа;
- Echo (R) – вывод сигнала выхода;
- GND – вывод «Земля».
- Размер 20 x 40 мм

### 3.4.2 Схема взаимодействия с Arduino

Для получения данных, необходимо выполнить такую последовательность действий (см. рис. 4):

- Подать на выход Trig импульс длительностью 10 микросекунд;
- В ультразвуковом дальнотере датчике HC – SR04 подключенном к Arduino произойдет преобразование сигнала в 8 импульсов с частотой 40 кГц, которые через излучатель будут посланы вперед;
- Когда импульсы дойдут до препятствия, они отразятся от него и будут приняты приемником, что обеспечит наличие входного сигнала на выходе Echo;
- На стороне контроллера полученный сигнал при помощи формул следует перевести в расстояние.

При делении ширины импульса на 58.2, получим данные в сантиметрах, при делении на 148 – в дюймах.

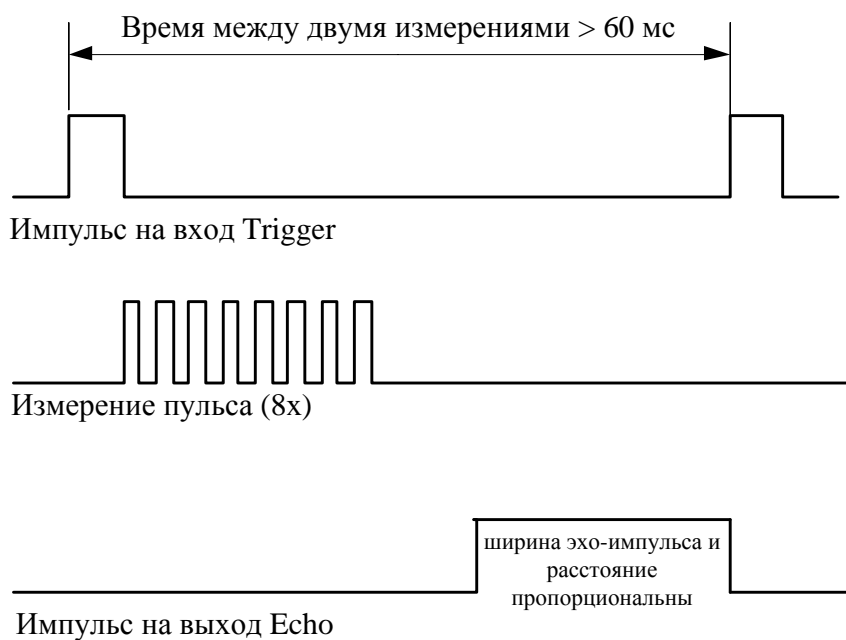


Рисунок 4. Принцип работы датчика.

Изн. № подл.	Подп. и дата
Взам. Изв. №	Изн. № дубл.
Подп. и дата	Подп. и дата

Изн.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)

### 3.4.3 Подключение датчика к Arduino

Подключение к плате Arduino ультразвукового датчика (см. рис. 2) выполняется очень просто. Контакты GND и 5V подключаются к соответствующим контактам на плате. Выходы Trig и Echo подсоединяем к Arduino на цифровые пины.

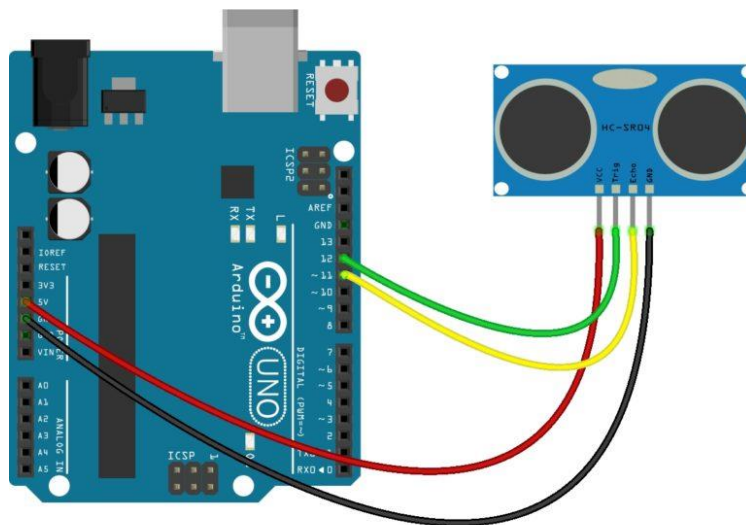


Рисунок 5. Схема подключения датчика к Arduino.

Для облегчения работы с датчиком будет использоваться библиотека Ultrasonic by Erick Simöns версии 3.0.0. Главным преимуществом данной библиотеки является удобство считывания показаний датчика и минимизация кода в основном теле программы.

Как и в любом устройстве измерения точность датчика зависит от нескольких факторов:

- температуры и влажности воздуха;
- расстояния до объекта;
- расположения относительно датчика (см. рис. 6);
- качества исполнения элементов модуля датчика.

В основу принципа действия любого ультразвукового датчика заложено явление отражения акустических волн, распространяющихся в воздухе. Но как известно из курса физики, скорость распространения звука в воздухе зависит от свойств этого самого воздуха (в первую очередь от температуры). Датчик же, испуская волны и замеряя время до их возврата, не догадывается, в какой именно среде они будут распространяться и берет для расчетов некоторую среднюю величину. В реальных условиях из-за фактора температуры воздуха датчик HC-SR04 может ошибаться от 1 до 3-5 см., что для разрабатываемой системы не является критичным.

Фактор расстояния до объекта важен, т.к. растет вероятность отражения от соседних

Инв. № подл.	Подп. и дата				Лист 16
	Инв. № дубл.				
	Взам. Инв. №				
	Подп. и дата				
ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)					
				Изм.	

Еrick Simõns версии 3.0.0. Главным преимуществом данной библиотеки является удобство считывания показаний датчика и минимизация кода в основном теле программы.

Как и в любом устройстве измерения точность датчика зависит от нескольких факторов:

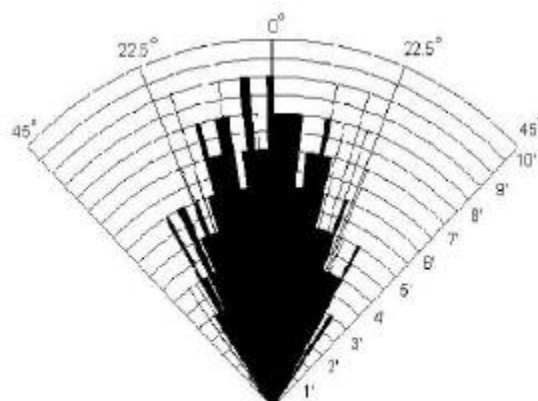
- температуры и влажности воздуха;
- расстояния до объекта;
- расположения относительно датчика (см. рис. 6);
- качества исполнения элементов модуля датчика.

В основу принципа действия любого ультразвукового датчика заложено явление отражения акустических волн, распространяющихся в воздухе. Но как известно из курса физики, скорость распространения звука в воздухе зависит от свойств этого самого воздуха (в первую очередь от температуры). Датчик же, испуская волны и замеряя время до их возврата, не догадывается, в какой именно среде они будут распространяться и берет для расчетов некоторую среднюю величину. В реальных условиях из-за фактора температуры воздуха датчик HC-SR04 может ошибаться от 1 до 3-5 см., что для разрабатываемой системы не является критичным.

Фактор расстояния до объекта важен, т.к. растет вероятность отражения от соседних

предметов, к тому же и сам сигнал затухает с расстоянием.

Так же для повышения точности надо правильно направить датчик: сделать так, чтобы предмет был в рамках конуса диаграммы направленности.



**практический тест производительности,  
лучше всего под углом 30 градусов**

Рисунок 6. Диаграмма рабочей области датчиков.

Излучатель и приёмник НС – SR04 должны смотреть прямо на предмет.

Для уменьшения ошибок и погрешности измерений обычно выполняются следующие действия:

- усредняются значения (несколько раз замеряем, убираем всплески, потом находим среднее);
- с помощью датчиков (например, DHT11 или DHT22) определяется температура и вносятся поправочные коэффициенты;
- датчик устанавливается на серводвигатель, для направления рабочей области влево или вправо.

Изн. № подл.	Подп. и дата	Взам. Изв. №	Изн. № дубл.	Подп. и дата	<div> <div>ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)</div> <div>Лист</div> <div>17</div> </div>				
Изн.	Лист	№ докум.	Подп.	Дата					

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Для робота телеприсутствия ElcBot в заданных режимах требуется 9 датчиков НС – SR04 расположенных по периметру робота. Данное расположение и количество (см. рис.7 и 8) является минимально необходимым, чтобы робот безошибочно определял наличие препятствий на протяжении всего периметра.



При движении вперёд препятствия могут иметь разный формат (например: стол со сплошной стороной, стул, человек и т.д.), для более точно определения препятствия один из датчиков расположен выше остальных с фронтальной стороны робота (см. рис. 8). В частности, данный датчик предназначен для определения на пути робота человека.



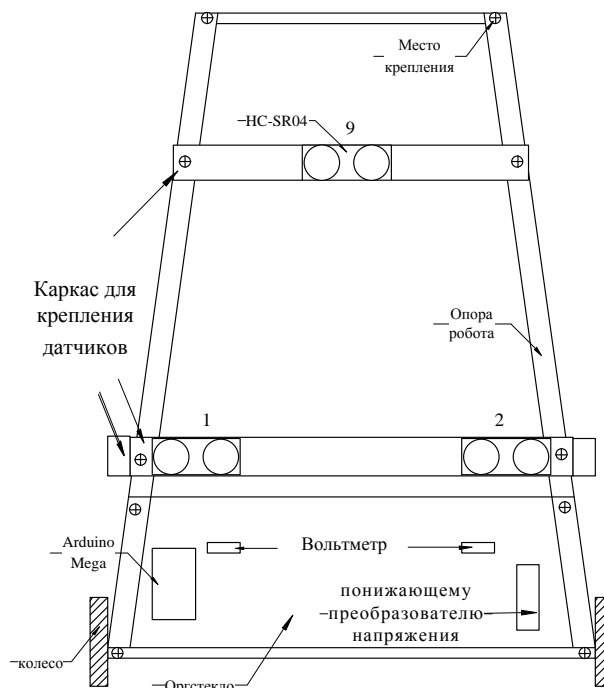


Рисунок 8. Схема расположения датчиков, вид спереди.

В качестве каркаса для крепления датчиков выбран алюминиевый профиль, это позволит жёстко зафиксировать датчики.

Для соединения датчиков с Arduino Mega использовались медные провода в несгораемой оплетке, но в качестве метола была выбрана пайка. Стандартные провода для подключения типа «папа – мама» не обеспечивают надёжное соединение без потери сигнала. Пайка проводов напрямую к платам обеспечивает надёжность конструкции во время движения робота. В результате воздействия вибрации на соединение стандартными проводами может произойти отсоединение, как следствие потеря сигнала и сбой в работе робота.

Ввиду большого количества датчиков требовалось решение проблемы подключения питания, земли и минимизации количества задействованных пинов платы Arduino Mega. Было принято решение об объединении однотипных контактов в один. Для снижения нагрузки на Arduino Mega 5V и GND подключили к понижающему преобразователю напряжения до 5V. Так как триггеры датчиков HC – SR04 используются для формирования сигнала и каждый датчик формирует его самостоятельно, то управляющий сигнал так же можно объединить и подключить к одному пину Arduino Mega. В дальнейшем это позволит подключить большее количество плат расширения, что позволит значительно расширить Функциональные возможности разрабатываемой системы в целом.

Проведя небольшое исследование, получили результат, что триггеры всех датчиков можно подключить не к цифровому порту, а к аналоговому.

Изн. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Изн.	Лист	№ докум.	Подп.	Дата

ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)

Лист
19

Аналоговые порты используются реже чем цифровые, и использование аналогового порта не приводит к функциональным неполадкам разрабатываемой системы, было принято решение использовать порт «А15».

Результатом сборки получилась следующая спецификация (см. рис. 9):

- 5V и GND датчиков 1 – 9 подключены к понижающему преобразователю напряжения до 5V;

- Trigger датчиков № 1 – 9 подключены к А15 Arduino Mega;

- Echo датчика № 1 – 27 Arduino Mega;

- Echo датчика № 2 – 35 Arduino Mega;

- Echo датчика № 3 – 37 Arduino Mega;

- Echo датчика № 4 – 31 Arduino Mega;

- Echo датчика № 5 – 29 Arduino Mega;

- Echo датчика № 6 – 33 Arduino Mega;

- Echo датчика № 7 – 25 Arduino Mega;

- Echo датчика № 8 – 23 Arduino Mega;

- Echo датчика № 9 – 39 Arduino Mega;

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата						Лист
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)					

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

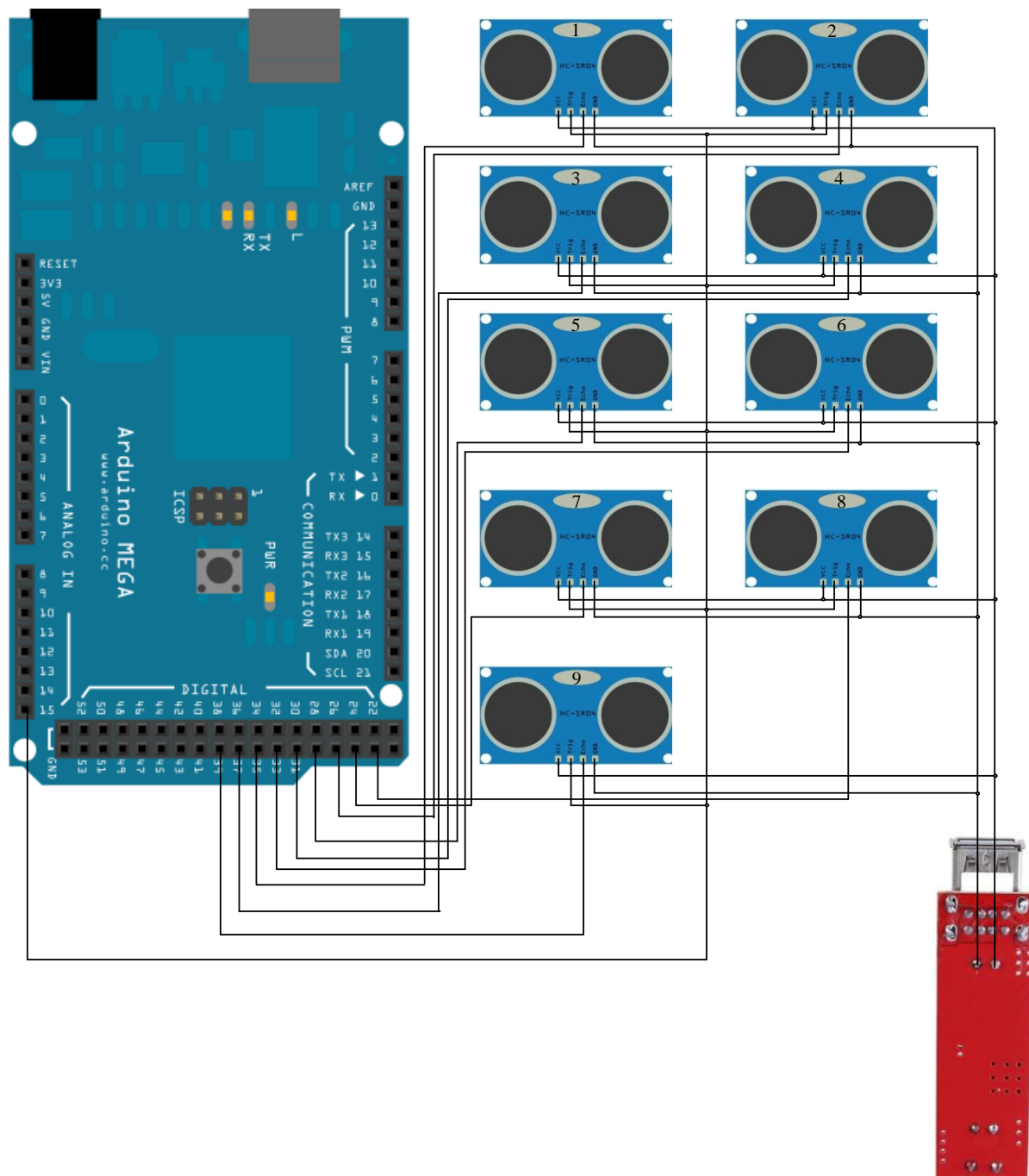


Рисунок 9. Структурная схема подключения датчиков к Arduino Mega.

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)

Лист

21

#### 4. Разработка алгоритмов

##### 4.1 Разработка алгоритма управления оператором

Рассмотрим блок схему работы робота в режиме управления оператором с компьютера (см. рис. 10). Оператор посылает управляющий сигнал по средствам подсистемы связи с Python 3 на Raspberry Pi. Сигнал распознаётся и код направляется в Arduino Mega, где выполняется основная программа.

В начале происходит опрос всех датчиков для определения дистанций до препятствия. В зависимости от управляющего сигнала происходит сравнение показаний нужных датчиков. При получении команды «W» движение вперёд в сравнении участвуют датчики 1, 2 и 9, «S» движении назад – датчики 5 и 6, с минимально – разрешённой дистанцией для выполнения команды. Минимально – разрешённая дистанция жёстко прописана в коде программы и меняется в зависимости от команды.

Результатом сравнения является выполнение управляющего сигнала, в случае если расстояние больше минимального, иначе управляющий сигнал принудительно меняется на запрещающий, соответственно выполнить движение в сторону препятствия не получится.

Для управляющих сигналов поворота «A» налево и «D» направо минимально разрешённая дистанция меньше чем для движения вперёд и назад, так как поворот робота происходит фактически на месте. При повороте в сравнении участвуют диагонально расположенные относительно друг друга датчики (см. рис. 7). Соответственно при команде «A» в сравнении с разрешённой дистанцией участвуют датчики 3 и 7, при команде «D» – датчики 4 и 8.

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)					Лист
										22



## 4.2 Разработка алгоритма обхода лабиринта

Одним из режимов робота является обход лабиринта в автоматическом режиме, но также данный режим может служить для обхода периметра в замкнутом помещении, в зависимости от задач робота.

Основным принципом данного режима является придерживание одной из сторон. В качестве таковой была выбрана правая сторона, т.е. робот всегда будет ехать вдоль правой стены и поворачивать направо при первой же возможности.

Для реализации данного режима будет использоваться модель конечного автомата, изображённого на рис. 11.

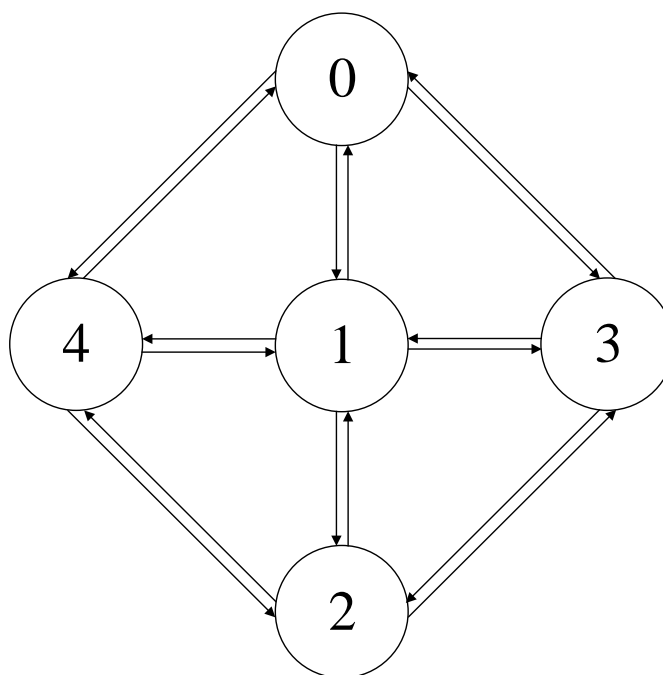


Рисунок 11 – Модель конечного автомата

Конечный автомат (или попросту FSM — Finite – state machine) это модель вычислений, основанная на гипотетической машине состояний. В один момент времени только одно состояние может быть активным. Следовательно, для выполнения каких-либо действий машина должна менять свое состояние.

Для включения режима «Обход лабиринта» в код программы управления необходимо встроить флаг сигнализирующий, по выставлению которого будет задействована модель конечного автомата. Из общей блок-схема (см. рис. 10) видно, что при получении команды «L» данный флаг будет выставлен в активное положение.

Рассмотрим подробнее все возможные состояния «Конечного автомата».

Состояние «0»: Препятствие спереди отсутствует. Датчик 8 определяет расстояние до препятствия меньше чем 60 см. На порты Arduino Mega подключённые к моторам ENA и

Инов. № подл.	Подп. и дата	Взам. Инов. №	Инов. № дубл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)

Лист
24

ENB подаются управляющие сигналы по изменению скорости вращения двигателей, чтобы робот начал движение таким образом, чтобы отъехать от препятствия на оптимальное расстояние;

Состояние «1»: Препятствие спереди отсутствует. Датчик 8 определяет расстояние до препятствия как оптимальное (больше 60 см, но меньше 70 см), выполняется движение вперёд;

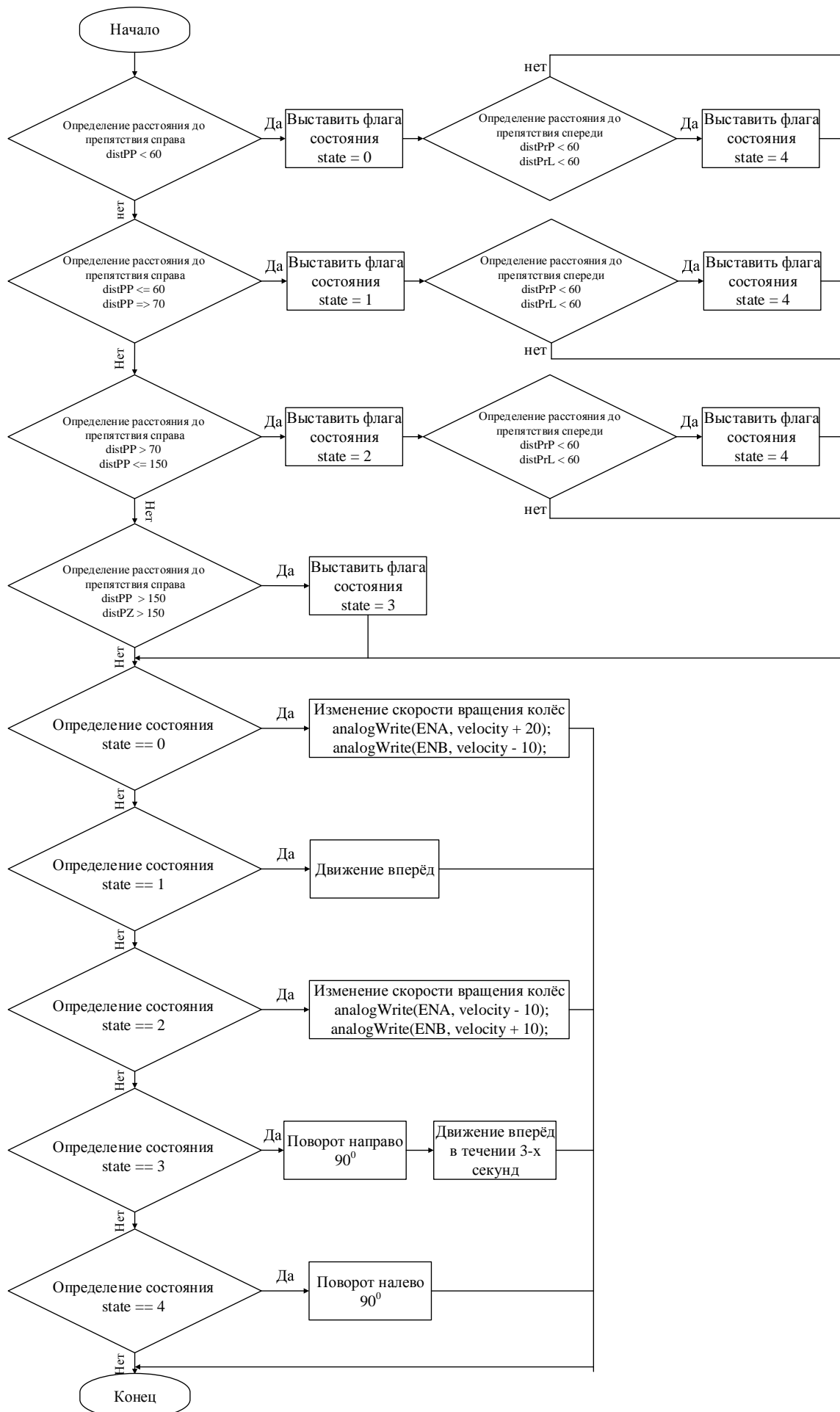
Состояние «2»: Препятствие спереди отсутствует. Датчик 8 определяет расстояние до препятствия больше 70 см., но меньше 150 см. На порты Arduino Mega подключённые к моторам ENA и ENB подаются управляющие сигналы по изменению скорости вращения двигателей, чтобы робот начал движение таким образом, чтобы приблизиться к препятствию справа на оптимальное расстояние;

Состояние «3»: Препятствие справа отсутствует. Датчики 7 и 8 определяют расстояние до препятствия больше 150 см. Робот выполняет поворот направо  $90^0$  и движение вперёд в течении 3 секунд. Это необходимо для того, чтобы робот после поворота определит препятствие с права и продолжил выполнение алгоритма.

Состояние «4»: Наличие препятствия спереди и справа. Один из датчиков 1, 2 или 9 определяют расстояние до препятствия меньше чем 60 см., а датчик 8 фиксирует наличие препятствия на расстоянии менее чем 150 см. Выполняет поворот налево  $90^0$ . В данном случаи принудительное движение вперёд не требуется, так как после поворота справа уже будет препятствие и выполнение движения продолжится в штатном режиме.

Исходя из описания возможных состояний конечного автомата блок –схема движения при выставлении влага «Обход лабиринта» будет выглядеть следующим образом:

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	Инв. № подл.					
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)					Лист
										25



Изн. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Изн.	Лист	№ докум.	Подп.	Дата

ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)



## 5.1 Разработка программных средств для управления оператором

Так как разрабатываемая система предназначена для готового робота все необходимые программные вставки для функционирования системы необходимо делать в уже готовом коде управления. Также все основные данные, скорость работы всех функциональных узлов заданы изначально.

Рассмотрим основные элементы кода для написания программы управления роботом:

```
#define SERIAL_SPEED 9600 – скорость работы последовательного порта;
```

```
#include "Ultrasonic.h" – подключение библиотеки для работы с датчиками HC – SR04;
```

int IN1 = 7; int IN2 = 6; int IN3 = 5; int IN4 = 4; – выходы Arduino Mega подключенные к драйверу двигателей для управления направлением вращения колёс;

```
int ENA = 9; int ENB = 3; – выходы Arduino Mega подключенные к моторам
колёс;
```

int distPrP – дистанция с датчика №1.

distPrL – дистанция с датчика №2,

distPP – дистанция с датчика №8,

distPZ – дистанция с датчика №7.

distLP– дистанция с датчика №3.

distLZ – дистанция с датчика №4.

distZP– дистанция с датчика №6,

distZL– дистанция с датчика №5,

distPrV – дистанция с датчика №9:

В соответствии с рис. 6 все датчики для работы с библиотекой необходимо указать следующие переменные

// Trig – A15, Echo - 46, 47, 48, 49, 50, 51, 52, 53.

Ultrasonic ultrasonic1(A15, 27);

Ultrasonic ultrasonic2(A15, 35);

Ultrasonic ultrasonic3(A15, 37);

Ultrasonic ultrasonic4(A15, 31);

Ultrasonic ultrasonic5(A15, 29);

Ultrasonic ultrasonic6(A15, 33);

Ultrasonic ultrasonic7(A15, 25);

Ultrasonic ultrasonic8(A15, 23);

Ultrasonic ultrasonic9(A15, 39);

Так же, как и датчики на рис. 6 ultrasonic 1 – 9 соответствуют датчикам по нумерации.  
const int stop\_distance = 60 – дистанция для сравнения с датчиками 1, 2, 5, 6 и 9 при движении вперёд и назад;

const int stop\_distance2 = 30 – дистанция для сравнения с датчиками 3, 4, 7, 8 используемых при поворотах;

С помощью команды ultrasonicN.read(), где N – это номер датчика, считываются показатели с нужных датчиков. При одновременном использовании близко расположенных датчиков нужно учитывать возможный сбой и разделить получение показателей по времени, используя задержку 50мс, необходимую для считывания показаний датчика, delay(50).

distPrV = ultrasonic9.read();

delay(50);

distPrL = ultrasonic2.read();

if (command == 'W') – определение управляющего сигнала;

{

if ((distPrP < stop\_distance) || (distPrL < stop\_distance) || (distPrV < stop\_distance)) – сравнение данных с датчиков с минимальной дистанцией для остановки робота

{

analogWrite(ENA, 0);

analogWrite(ENB, 0) – команды для управления двигателями колёс;

command = 'x' – смена управляющего сигнала;

}

}

digitalWrite (IN2, LOW);

digitalWrite (IN1, HIGH);

digitalWrite (IN4, LOW);

digitalWrite (IN3, HIGH); команды для управления направлением вращения колёс.

switch (command) {

case 'W': – блок кода отвечающий за распознавание управляющего сигнала.

Изн. № подл.	Подп. и дата	Взам. Изв. №	Изн. № дубл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)

Лист
28

## 5.2 Разработка программных средств для обхода лабиринта

Основным отличием в режимах являются способы формирования управляющих сигналов, передаваемых на моторы и драйвер двигателя. При управлении роботом в ручном режиме сигналы формируются по средствам передачи сигналов управления с компьютера.

В случае переключения режима меняется метод формирования управляющих сигналов. Единственным управляющим сигналом, поступающим на Arduino Mega с подсистемы управления, является команда «L», поэтому формирование сигналов управления двигателем должно происходить автоматически, в зависимости от показаний датчиков системы распознавания препятствий.

Основные переменные остаются теми же, что и в режиме управления оператором. Поэтому рассмотрим команды для автономного движения робота:

```
if (command != 'L')
{
    labflag = -1;
    state = -1;
}
```

Данный блок кода отвечает за отключение режима обхода лабиринта и выставление нейтрального состояния системы

«Конечный автомат» представляет из себя набор условий, состоящий из двух основных блоков. Первый блок основываясь на полученных данных от внешних устройств, задаёт нужное значение переменной «state», отвечающей за состояние системы движения. Соответственно данные, о положении робота относительно окружения, поступают от системы определения препятствий, основанная на датчиках HC – SR04:

```
if (distPP < 60)
    state = 0;
if ((distPP >= 60) && (distPP <= 70))
    state = 1;
if ((distPP > 70) && (distPP <= 150))
    state = 2;
if ((distPP > 150) && (distPZ > 150))
    state = 3;
if ((distPrP < 60) || (distPrL < 60) || (distPrV < 60))
    state = 4;
```

Изн. № подл.	Подп. и дата	Взам. Изв. №	Изн. № дубл.	Подп. и дата	<div>ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)</div>					Лист
										29
Изм.	Лист	№ докум.	Подп.	Дата						

После этого второй блок «Конечного автомата» формирует сигналы управления моторами и драйвером двигателя, в соответствии с выставленным состоянием системы.

```

if (labflag == 1)
{
    // Turn Left
    if (state == 0)
    {...}
    // Drive
    if (state == 1)
    {...}
    // Turn Right
    if (state == 2)
    {...}
    // To Right
    if (state == 3)
    {...}
    // To Left
    if (state == 4)
    {...}

```

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)					Лист
										30

## 6. Тестирование разрабатываемой системы

Тестирование системы является одной из самых важных частей процесса разработки и проверки полученного результата. В процессе разработки тестирование позволяет выявить недостатки как программной, так и аппаратной частей:

- Программной части: Позволяет выявить, на начальной стадии, правильный выбор алгоритма и его реализация в программном коде. Выявлять и исправлять ошибки, возникающие в процессе написания основной части, которые невозможно выявить без фактического испытания или его моделирования в виртуальном пространстве.

- Аппаратной части: Также важная часть разработки. В рамках выпускной квалификационной работы это позволило выявить недостатки изначального размещения датчиков по периметру робота, нехватку датчиков для нормального функционирования системы. Относительно начальных требований к разработке количество датчиков увеличилось в два раза, и поменялось расположение датчиков на роботе.

Проверка работоспособности системы проводилась с начала разработки, и финальным этапом разработки является фактическое подтверждение заявленных требований системы. Подтверждением результата разработки программно-аппаратной системы обнаружения препятствий для мобильного робота является видеоматериал, приложенный к данной работе.

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)					Лист
										31

### Заключение

В результате выполнения выпускной квалификационной работы была разработана программно-аппаратная система обнаружения препятствий для мобильного робота телеприсутствия ElcBot. Система основана на эффекте эхолокации.

Система стабильно и безошибочно распознаёт препятствие и не позволяет врезаться в препятствия при движении в различных режимах. Результатом явилось внедрение разрабатываемой системы в проект робота телеприсутствия ElcBot.

В дальнейшем данную систему можно будет модернизировать и улучшить быстродействие и расширить функциональность системы в целом.

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	Изм.	Лист	№ докум.	Подп.	Дата	ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)	Лист
											32

# Список литературы

1. Сайт проекта робота телеприсутствия [Электронный ресурс]. Режим доступа: <https://tinyroboticsteam.github.io>
2. Википедия – Свободная энциклопедия [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/>
3. Основы робототехники и общие сведения о роботах. Устройство роботов [Электронный ресурс]. Режим доступа: <https://www.raspberrypi.org/>
4. Материалы по программированию Arduino [Электронный ресурс]. Режим доступа: <https://arduino.ru>
5. ARDUINOMASTER Русское Ардуино – сообщество [Электронный ресурс]. Режим доступа: <https://arduinomaster.ru>
6. RobotChip – Блог о проектах Arduino, библиотеки, обзоры [Электронный ресурс]. Режим доступа: <https://robotchip.ru>
7. Raspberry Pi – Teach, Learn and Make with Raspberry Pi [Электронный ресурс]. Режим доступа: <https://www.raspberrypi.org/>
8. Сообщество EasyElectronics.ru [Электронный ресурс]. Режим доступа: <http://we.easyelectronics.ru/>
9. Pikabu русскоязычное информационно – развлекательное сообщество [Электронный ресурс]. Режим доступа: <https://pikabu.ru/>

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)					Лист
										33

## Приложения

Приложение 1. Исходный код исполнительской части.

```
#define SERIAL_SPEED 9600
```

```
#include "Ultrasonic.h"
```

```
#define READ_SENSOR_INTERVAL 100UL
```

```
int IN1 = 7; // Input1 подключен к выводу 7
```

```
int IN2 = 6;
```

```
int IN3 = 5;
```

```
int IN4 = 4;
```

```
int ENA = 9;
```

```
int ENB = 3;
```

```
char command = 'S';
```

```
char prevCommand = 'A';
```

```
int velocity = (4 + 1) * 10 + 100;
```

```
unsigned long timer0 = 2000;
```

```
unsigned long timer1 = 0;
```

```
int distPrP, distPrL, distPP, distPZ, distLP;
```

```
int distLZ, distZP, distZL, distPrV;
```

```
// Trig - A5,
```

```
// Echo - 46, 47, 48, 49, 50, 51, 52, 53.
```

```
Ultrasonic ultrasonic1(A15, 27);
```

```
Ultrasonic ultrasonic2(A15, 35);
```

```
Ultrasonic ultrasonic3(A15, 37);
```

```
Ultrasonic ultrasonic4(A15, 31);
```

```
Ultrasonic ultrasonic5(A15, 29);
```

```
Ultrasonic ultrasonic6(A15, 33);
```

```
Ultrasonic ultrasonic7(A15, 25);
```

```
Ultrasonic ultrasonic8(A15, 23);
```

```
Ultrasonic ultrasonic9(A15, 39);
```

```
const int stop_distance = 60; //cm
```

Изм.	Лист	№ докум.	Подп.	Дата	ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)	Лист
Взам. Инв. №	Инв. № дубл.	Подп. и дата	Подп. и дата	Инв. № подл.		34



```
const int stop_distance2 = 30; //cm
```

```
int state = -1;
```

```
void setup() {
```

```
    Serial.begin(SERIAL_SPEED);
```

```
    pinMode (ENA, OUTPUT);
```

```
    pinMode (IN1, OUTPUT);
```

```
    pinMode (IN2, OUTPUT);
```

```
    pinMode (ENB, OUTPUT);
```

```
    pinMode (IN4, OUTPUT);
```

```
    pinMode (IN3, OUTPUT);
```

```
}
```

```
int labflag = -1;
```

```
static unsigned long prevSensorTime = 0;
```

```
void loop()
```

```
{
```

```
    if ((unsigned long)(millis() - prevSensorTime) > READ_SENSOR_INTERVAL)
```

```
    {
```

```
        distPrP = ultrasonic1.read();
```

```
        distLP = ultrasonic3.read();
```

```
        distZL = ultrasonic5.read();
```

```
        distPZ = ultrasonic7.read();
```

```
        distPrV = ultrasonic9.read();
```

```
        delay(50);
```

```
        distPrL = ultrasonic2.read();
```

```
        distLZ = ultrasonic4.read();
```

```
        distPrP = ultrasonic6.read();
```

```
        distPP = ultrasonic8.read();
```

```
        if (command != 'L')
```

```
        {
```

```
            labflag = -1;
```

```
            state = -1;
```

Изн. № подл.	Подп. и дата	Взам. Изн. №	Изн. № дубл.	Подп. и дата

Изн. № подл.	Подп. и дата	Взам. Изн. №	Изн. № дубл.	Подп. и дата

БКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)

Лист

35

```

}
if (distPP < 60)
    state = 0;
if ((distPP >= 60) && (distPP <= 70))
    state = 1;
if ((distPP > 70) && (distPP <= 150))
    state = 2;
if ((distPP > 150) && (distPZ > 150))
    state = 3;
if ((distPrP < 60) || (distPrL < 60) || (distPrV < 60))
    state = 4;

```

```

if (labflag == 1)
{
    if (state == 0)
    {
        digitalWrite(IN2, LOW);
        digitalWrite(IN1, HIGH);
        digitalWrite(IN4, LOW);
        digitalWrite(IN3, HIGH);
        analogWrite(ENA, velocity + 20);
        analogWrite(ENB, velocity - 10);
    }

```

```

    if (state == 1)
    {
        analogWrite(ENA, 0);
        analogWrite(ENB, 0);
        delay(20);
        // Смена направления вращения колёс
        digitalWrite(IN2, LOW);
        digitalWrite(IN1, HIGH);
        digitalWrite(IN4, LOW);
        digitalWrite(IN3, HIGH);
        // Установка скважности ШИМ
        analogWrite(ENA, velocity);
    }

```

Изн. № подл.	Подп. и дата	Изн. № дубл.	Взам. Изм. №	Подп. и дата	Изн. № подл.																							
Изн. № подл.	Подп. и дата	Изн. № дубл.	Взам. Изм. №	Подп. и дата	Изн. № подл.																							
Изн. № подл.	Подп. и дата	Изн. № дубл.	Взам. Изм. №	Подп. и дата	Изн. № подл.																							
Изн. № подл.	Подп. и дата	Изн. № дубл.	Взам. Изм. №	Подп. и дата	Изн. № подл.																							
Изн. № подл.	Подп. и дата	Изн. № дубл.	Взам. Изм. №	Подп. и дата	Изн. № подл.																							
Изн. № подл.	Подп. и дата	Изн. № дубл.	Взам. Изм. №	Подп. и дата	Изн. № подл.																							
Изн. № подл.	Подп. и дата	Изн. № дубл.	Взам. Изм. №	Подп. и дата	Изн. № подл.																							

ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)

Лист

36

```

    analogWrite(ENB, velocity);
}

if (state == 2)
{
    // В - левое колесо
    // А - правое колесо
    // больше скорости на правое
    digitalWrite (IN2, LOW);
    digitalWrite (IN1, HIGH);
    digitalWrite (IN4, LOW);
    digitalWrite (IN3, HIGH);
    analogWrite(ENA, velocity - 10);
    analogWrite(ENB, velocity + 10);
}

if (state == 3)
{
    // остановка
    analogWrite(ENA, 0);
    analogWrite(ENB, 0);
    delay(20);

    digitalWrite (IN2, LOW);
    digitalWrite (IN1, HIGH);
    digitalWrite (IN4, LOW);
    digitalWrite (IN3, HIGH);
    analogWrite(ENA, velocity);
    analogWrite(ENB, velocity);

    delay(1000);

    // поворот через delay
    // А
    digitalWrite (IN2, HIGH);
    digitalWrite (IN1, LOW);
    // В

```

Изн. № подл.	Подп. и дата	Взам. Изв. №	Изн. № дубл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)

```

digitalWrite (IN4, LOW);
digitalWrite (IN3, HIGH);

analogWrite(ENA, velocity);
analogWrite(ENB, velocity);
delay(800);

digitalWrite (IN2, LOW);
digitalWrite (IN1, HIGH);
digitalWrite (IN4, LOW);
digitalWrite (IN3, HIGH);
analogWrite(ENA, velocity);
analogWrite(ENB, velocity);

delay(3000);
// проезд вперёд

analogWrite(ENA, 0);
analogWrite(ENB, 0);

//delay(50000);
}
if (state == 4)
{
    analogWrite(ENA, 0);
    analogWrite(ENB, 0);
    delay(20);

    // поворот налево через
    // A
    digitalWrite (IN2, LOW);
    digitalWrite (IN1, HIGH);
    // B
    digitalWrite (IN4, HIGH);
    digitalWrite (IN3, LOW);

```

Изн. № подл.	Подп. и дата	Взам. Изн. №	Изн. № дубл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)

delay(800);

digitalWrite (IN2, LOW);

digitalWrite (IN1, HIGH);

digitalWrite (IN4, LOW);

digitalWrite (IN3, HIGH);

analogWrite (ENA, velocity);

analogWrite (ENB, velocity);

delay(2000);

analogWrite(ENA, velocity);

analogWrite(ENB, velocity);

delay(800);

analogWrite(ENA, 0);

analogWrite(ENB, 0);

}

}

prevSensorTime = millis();

if (command == 'W'){

if ((distPrP < stop\_distance) || (distPrL < stop\_distance) || (distPrV < stop\_distance))

{

analogWrite(ENA, 0);

analogWrite(ENB, 0);

command = 'x';

}

}

if (command == 'A'){

if ((distLP < stop\_distance2) || (distPZ < stop\_distance2))

{

analogWrite(ENA, 0);

analogWrite(ENB, 0);

command = 'x';

}

Изн. № подл.	Подп. и дата	Взам. Изн. №	Изн. № дубл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-ИГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)

Лист
39

```

    }
    if (command == 'S'){
        if ((distZP < stop_distance) || (distZL < stop_distance))
        {
            analogWrite(ENA, 0);
            analogWrite(ENB, 0);
            command = 'x';
        }
    }
    if (command == 'D'){
        if ((distLZ < stop_distance2) || (distPP < stop_distance2))
        {
            analogWrite(ENA, 0);
            analogWrite(ENB, 0);
            command = 'x';
        }
    }
}

```

```

if (Serial.available() > 0) {
    timer1 = millis();
    prevCommand = command;
    command = Serial.read();
}

```

```

if (command != prevCommand) {
    switch (command) {
        case 'W':
            analogWrite(ENA, 0);
            analogWrite(ENB, 0);
            delay(20);
            digitalWrite (IN2, LOW);
            digitalWrite (IN1, HIGH);
            digitalWrite (IN4, LOW);
            digitalWrite (IN3, HIGH);
            // Установка скважности ШИМ
            analogWrite(ENA, velocity);

```

Изн. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)					Лист
										40
Изм.	Лист	№ докум.	Подп.	Дата						

```

    analogWrite(ENB, velocity);

    break;
case 'A':
    analogWrite(ENA, 0);
    analogWrite(ENB, 0);
    delay(20);
    digitalWrite (IN2, LOW);
    digitalWrite (IN1, HIGH);
    digitalWrite (IN4, HIGH);
    digitalWrite (IN3, LOW);

    analogWrite(ENA, velocity);
    analogWrite(ENB, velocity);

    break;
case 'S':
    analogWrite(ENA, 0);
    analogWrite(ENB, 0);
    delay(20);

    digitalWrite (IN2, HIGH);
    digitalWrite (IN1, LOW);
    digitalWrite (IN4, HIGH);
    digitalWrite (IN3, LOW);

    analogWrite(ENA, velocity);
    analogWrite(ENB, velocity);

    break;
case 'D':
    labflag = 2;

    analogWrite(ENA, 0);
    analogWrite(ENB, 0);
    delay(20);
    digitalWrite (IN2, HIGH);

```

Изн. № подл.	Подп. и дата	Взам. Изв. №	Изн. № дубл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)

```
digitalWrite (IN1, LOW);  
digitalWrite (IN4, LOW);  
digitalWrite (IN3, HIGH);
```

```

    analogWrite(ENA, velocity);
    analogWrite(ENB, velocity);
    break;
case ' ':
    analogWrite(ENA, 0);
    analogWrite(ENB, 0);
    break;

```

```
case 'L':
    //velocity = 0;
    if (labflag != 1)
        labflag = 1;
    break;
```

```
default:
    if ((command >= 48) && (command <= 57)) {
        if (command == 48)
        {
            velocity = 0;
        }
        else
        {
            velocity = (command - 48 + 1) * 10 + 100;
        }
    }
}
```

```
}  
}  
  
else {  
    timer0 = millis(); // Получение текущего времени  
    if ((unsigned long)(timer0 - timer1) > 20000) {  
        analogWrite(ENA, 0);
```

[illegible]



```

analogWrite(ENB, 0);
prevCommand = '';
labflag = -1;
}
}
}

```

Инв. № подл.	Подп. и дата			
	Инв. № дубл.			
	Взам. Инв. №			
Инв. № подл.	Подп. и дата			
	Инв. № дубл.			
	Взам. Инв. №			

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)

Лист
43

Приложение 2. Исходный код клиентской части.

```

from __future__ import print_function, absolute_import, division
import subprocess
import urwid
import signal
import os
import serial
from subprocess import Popen, PIPE
from time import sleep
def exit_on_q(key):
    global power
    global ser
    global spower
    global p
    global currc

    if key in ('q', 'Q', 'й', 'Й'):
        #if ser != -1:
        #ser.write(b'q')
        p.stdin.write(b'Q\n')
        p.stdin.flush()
        sleep(1)
        # https://stackoverflow.com/questions/4789837/how-to-terminate-a-python-subprocess-launched-with-shell-true
        os.killpg(os.getpgid(p.pid), signal.SIGTERM)
        raise urwid.ExitMainLoop()

    #if ser == -1:
    #    return
    if key in ('w', 'W', 'и', 'И'):
        # forward
        #ser.write(b'W')#pw = str(power)
        currc = 'W - Forward'
        #string1 = 'W'
        #string1_encode = string1.encode()
        #ser.write(string1_encode)#pw = str(power)
        p.stdin.write(b'W\n')

```

Изм. Лист	№ докум.	Подп.	Дата	Инв. № подл.	Подп. и дата	<div>#ser.write(b'q') p.stdin.write(b'Q\n') p.stdin.flush() sleep(1) # <a href="https://stackoverflow.com/questions/4789837/how-to-terminate-a-python-subprocess-launched-with-shell-true">https://stackoverflow.com/questions/4789837/how-to-terminate-a-python-subprocess-launched-with-shell-true</a> os.killpg(os.getpgid(p.pid), signal.SIGTERM) raise urwid.ExitMainLoop() #if ser == -1: # return if key in ('w', 'W','u','U'): # forward  #ser.write(b'W')#pw = str(power) currс = 'W - Forward' #string1 = 'W' #string1_encode = string1.encode() #ser.write(string1_encode)#pw = str(power) p.stdin.write(b'W\n')</div>
					Взам. Инв. №	
					Инв. № дубл.	
					Подп. и дата	
					Подп. и дата	
BKP-ИГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)						
Лист						
44						

```

p.stdin.flush()
if key in ('a', 'A', 'ф', 'Ф'):
# Left
#ser.write(b'A')#pw = str(power)
currс = 'A - Left'
#pw = str(power)
#txt_CP.set_text(('banner', u"A"))
#power = 30
#spower = 3
#txt_CP.set_text(('banner', str(power)))
p.stdin.write(b'A\n')
p.stdin.flush()
if key in ('s', 'S', 'ы', 'Ы'):
# Backward
#ser.write(b'S')#pw = str(power)
currс = 'S - Backward'
#pw = str(power)
#txt_CP.set_text(('banner', u"S"))
p.stdin.write(b'S\n')
p.stdin.flush()
if key in ('d', 'D', 'в', 'В'):
# Right
#ser.write(b'D')
currс = 'D - Right'
#txt_CP.set_text(('banner', u"D"))
#power = 30
#spower = 3
#txt_CP.set_text(('banner', str(power)))
p.stdin.write(b'D\n')
p.stdin.flush()
if key in ('l', 'L'):
# Labirint
currс = 'L - Labirint'
p.stdin.write(b'L\n')
p.stdin.flush()

```

Изн. № подл.	Подп. и дата	Изн. № дубл.	Подп. и дата	Взам. Изн. №

Изм.	Лист	№ докум.	Подп.	Дата

ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)

```

if key in (' '):
# Stop
    #ser.write(b' ');
    currc = 'Space - Stop'
    #power = 0
    #spower = 0
    #txt_CP.set_text(('banner', str(power)))
    #txt_CP.set_text(('banner', u"Space"))
    p.stdin.write(b' \n')
    p.stdin.flush()
if key in ('+'):
    if (power < 99):
        power = power + 10
        spower = spower + 1
        txt_CP.set_text(('banner', str(power)))
        p.stdin.write(bytes([spower+48]) + b'\n')
        p.stdin.flush()
        #ser.write(bytes([spower+48]))
if key in ('-'):
    if (power > 0):
        power = power - 10
        spower = spower - 1
        txt_CP.set_text(('banner', str(power)))
        p.stdin.write(bytes([spower+48]) + b'\n')
        p.stdin.flush()
        #ser.write(bytes([spower+48]))
txt_CCV.set_text(('banner', currc))
def enter_idle():
    loop.remove_watch_file(pipe.stdout)
def update_text(read_data):
    txt_Q.set_text(('banner', read_data))
# mlog = open("RoboLog.txt", "a+")
# mlog.write(str(read_data))
# mlog.write("\n")
# mlog.close()

```

Изн. № подл.	Подп. и дата	Взам. Изн. №	Изн. № дубл.	Подп. и дата

Изн.	Лист	№ докум.	Подп.	Дата

ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)

```

if __name__ == '__main__':
    currc = "No command"
    palette = [
        ('banner', 'black', 'light gray'),
        ('streak', 'black', 'dark blue'),
        ('bg', 'black', 'dark blue'),]
    # spower = 0..9 (48 .. 57)
    spower = 4
    power = spower * 10
    txt_F = urwid.Text(('banner', u"W - Forward (\u2191)"), align='center')
    txt_LRS = urwid.Text(('banner', u"\u2190 A - Left | Space - Stop | D - Right \u2192"),
align='center')
    txt_B = urwid.Text(('banner', u"S - Backward (\u2193)"), align='center')
    txt_P = urwid.Text(('banner', u"' + ' Increase motor power | ' - ' Decrease motor power"),
align='center')
    txt_C = urwid.Text(('banner', u"Current power:"), align='center')
    txt_CP = urwid.Text(('banner', str(power)), align='center')
    # current command
    txt_CC = urwid.Text(('banner', u"Current command: "), align='center')
    txt_CCV = urwid.Text(('banner', u"No command"), align='center')
    txt_Log = urwid.Text(('banner', u"Log: "), align='center')
    txt_LogV = urwid.Text(('banner', u""), align='center')
    txt_Q = urwid.Text(('banner', u"Q - Quit"), align='center')
    #txt_F = urwid.Text(('banner', u"W \u2191"), align='center')
    #txt_LRS = urwid.Text(('banner', u"\u2190 A | Space - Stop | D \u2192"), align='center')
    #txt_B = urwid.Text(('banner', u"S \u2193"), align='center')
    #empty string
    txt_E = urwid.Text(('banner', u""), align='center')
    pile = urwid.Pile([txt_F, txt_LRS, txt_B, txt_E, txt_P, txt_C, txt_CP, txt_E, txt_CC, txt_CCV,
txt_E, txt_Log, txt_LogV, txt_E, txt_Q ])
    top = urwid.Filler(pile, top = 5)
    #ser = -1
    #try:
    #    ser = serial.Serial('/dev/ttyACM0', 9600)
    #except serial.serialutil.SerialException:
    #    txt_LogV.set_text(('banner', '[-ERR] Could not connect to Arduino'))

```

Ив. № подл.	Подп. и дата	Взам. Ив. №	Ив. № дубл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата

БКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)

```

# if ser != -1:
#     txt_LogV.set_text(('banner', '[+OK] Connected to Arduino'))
loop = urwid.MainLoop(top, palette, unhandled_input=exit_on_q, handle_mouse=False)
    stdout = loop.watch_pipe(update_text)
    stderr = loop.watch_pipe(update_text)
    # pipe = subprocess.Popen('for i in $(seq 50); do echo -n "$i "; sleep 0.5; done', shell=True,
stdout=stdout, stderr=stderr)
    p = subprocess.Popen(['python3', 'shell_edt.py'], stdin = PIPE, stdout = stdout, stderr = stdout,
shell = False)
    loop.run()

```

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ докум.	Подп.	Дата	ВКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)					Лист
										48

Приложение 3. Исходный код управляющего блока.

```

serialport = serial.Serial(devName, serialspeed, timeout = 0.2)
import threading
import serial
from time import sleep
import sys
import threading
import serial
from time import sleep

global currcomm

#writetimer = 1 #

readtimer = 0.25 #

# rwflag, 0 - can read, 1 - can write
def read():
    global serialport
    global currcomm

    threading.Timer(readtimer, read).start()

    if (currcomm != -1):
        data = serialport.read(11);
        if (len(data) > 0):
            print(str(data) + " : " + str(len(data)))
            sys.stdout.flush();
#serialport = serial.Serial("/dev/ttyAMA0", 9600, timeout=0.5)

serialport = -1

print('Please wait ...');
sys.stdout.flush();

serialspeed = 38400

```

Инв. № подл.	Подп. и дата	Инв. № дубл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата
Инв. № подл.	Подп. и дата	Инв. № дубл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата
Инв. № подл.	Подп. и дата	Инв. № дубл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата
Инв. № подл.	Подп. и дата	Инв. № дубл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата	Инв. №	Подп. и дата

```

ind = range(0,10)
for i in ind:
    devName = '/dev/ttyUSB' + str(i)
    try:
        serialport = serial.Serial(devName, serialspeed, timeout = 0.2)
        sleep(3);
    except serial.serialutil.SerialException:
        #txt_LogV.set_text(('banner', '[-ERR] Could not connect to Arduino'))
        print('[-ERR] Could not connect to Arduino');
        sys.stdout.flush();
    print('Please wait ...');
    sys.stdout.flush();
    if serialport != -1:
        print('[+OK] Connected to Arduino, dev = ' + devName);
        sys.stdout.flush();
        break;
currcomm = -1

```

```

threading.Timer(readtimer, read).start()
sleep(1)

```

```

powerlevels = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10']
commands = ['W', 'A', 'S', 'D', ' ' ]
while True:
    currcomm = input()

    #print(currcomm) #
    #sys.stdout.flush();
    if (currcomm in powerlevels):
        serialport.write(bytes(currcomm, encoding = 'utf-8'));

    if (currcomm in commands):
        serialport.write(bytes(currcomm, encoding = 'utf-8'));

    if (currcomm == 'Q'):
        serialport.close() # Only executes once the loop exits

```

Изн. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата

БКР-НГТУ-09.03.01-(14-ВМ)-003-2019(ПЗ)

Лист

50