

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра «Системы автоматики, автоматизированное управление и
проектирование»

УТВЕРЖДАЮ
Заведующий кафедрой
_____ С.В. Ченцов
« ____ » _____ 2021 г.

БАКАЛАВРСКАЯ РАБОТА

27.03.04 «Управление в технических системах»

**СИСТЕМА ПРЕДОТВРАЩЕНИЯ СТОЛКНОВЕНИЙ
ЭЛЕКТРОСАМОКАТА**

Руководитель	_____	___06. 2021 г.	ст. преподаватель П. В. Авласко
Консультант	_____	___06. 2021 г.	С. В. Ченцов
Выпускник	_____	___06. 2021 г.	А. А. Огородников
Нормоконтролер	_____	___06. 2021 г.	Т. А. Грудинова

Красноярск 2021

РЕФЕРАТ

Выпускная квалификационная работа на тему "Система предотвращения столкновений электросамоката" содержит 42 страниц текстового документа, 1 приложение, 20 использованных источников.

СИСТЕМА СО ВСТРОЕННЫМ УЛЬТРАЗВУКОМ ДАТЧИКОМ

Цель работы

Изучение и разработка системы предотвращения столкновений электросамоката.

Задачи, которые решались в ходе выполнения данной бакалаврской работы:

- изучение имеющиеся системы регулировки скорости;
- изучение схемы подключения систем;
- разработать(выбрать) подходящую систему;
- реализовать схему со встроенным датчиком.

При выполнении данной работы был произведен обзор предметной области, задания на выпускную квалификационную работу, изучены существующие аналоги и сформированы требования, предъявляемые к системам регулировки скорости.

В результате была разработана система, реализующая поставленные задачи.

СОДЕРЖАНИЕ

Введение.....	5
1 Система предотвращения столкновений электросамоката.....	7
1.1 ТАУ.....	7
1.1.1 Управление от ЭВМ.....	7
1.1.2 Автоматическое регулирование	9
1.1.3 Обобщенная САУ.....	10
1.2 Программирование.....	13
1.2.1 C++.....	13
1.2.2 Java.....	14
1.2.3 Python.....	15
1.2.4 Arduino.....	17
2 Примеры датчиков	19
2.1 САРД G259	19
2.1.1 Описание САРД G259.....	19
2.1.2 Усилитель тормозного привода с электронным управлением.....	20
2.1.3 Выключатель САРД.....	21
2.1.4 Давление в тормозной магистрали.....	21
2.1.5 Обмен данными через шину CAN.....	22
2.1.6 Функциональные возможности системы.....	24
2.2 Система с лазерным датчиком.....	26
2.2.1 Lidar	26
2.2.2 Излучатель	29
2.2.3 Сканирующая оптика.....	30
3 Система с ультразвуковым датчиком	31
3.1 Описание	31
3.2 Arduino Uno.....	33
Заключение	36
Список сокращений	37

Список использованных источников	38
Приложение А	40

ВВЕДЕНИЕ

Безопасность участников движения на дороге – важнейший фактор, которому уделяют внимание большинство производителей транспортных средств (ТС). Система автоматического торможения является одним из основных элементов системы безопасности ТС для предотвращения столкновений. Она является чрезвычайно важным ассистентом, поскольку способна в одно мгновение оценить сложность возникшей на дороге ситуации и принять соответствующее решение об ее исправлении. Наверняка, многим водителям знакома ситуация, когда перед ТС на дороге резко появляется пешеход или какое-либо иное препятствие, способное спровоцировать ДТП с непредсказуемыми последствиями. Именно в таких случаях ассистент, позволяющий предотвратить столкновение, является незаменимым. Каждый производитель по-своему реализует данный сервис, однако принцип его действия абсолютно одинаков во всех случаях. Наличие системы автоторможения, способной предотвратить столкновения, является чрезвычайно важным моментом. Основной задачей подобного сервиса является оценка ситуации на дороге, распознавание всех объектов, дорожной разметки, предупреждение водителя о том, что расстояние между ТС и другим объектом резко сокращается и может привести к негативным последствиям посредством различных световых и звуковых сигналов. Если же водитель никаким образом не отреагировал на данное уведомление, то ассистент способен самостоятельно принять решение об активации автоторможения, чтобы избежать ДТП. Особенно важным подобный сервис является в том случае, когда непредвиденная ситуация возникла спонтанно и у водителя просто не остается времени на реальную, трезвую оценку положения дел на дороге. Несмотря на то, что данный сервис является достаточно новым и молодым, его постоянное совершенствование и способность помочь предотвратить аварию либо минимизировать ее

последствия, является чрезвычайно важным. Как правило, система автоторможения может помочь в таких случаях:

- водитель потерял концентрацию за рулем и не успел отреагировать на возникновение внештатной ситуации;
- на дороге очень плохая видимость (туман, слепящее солнце, метель, дождь);
- неожиданная ситуация в следствии появлении пешехода или иного препятствия перед ТС, либо резкое торможение находящегося впереди автотранспорта.

1 Система предотвращения столкновений электросамоката

Нужно создать систему, реализующую последующий перечень возможностей:

- слежение за движущимися впереди объектами;
- предупреждение о сближении с передвигающимися(покоящимся) впереди объектом;
- если необходимо, активация тормоза;
- автоматическая регулировка скорости по заданным данным.

Обобщенная схема реализации системы приведена на рисунке 1.

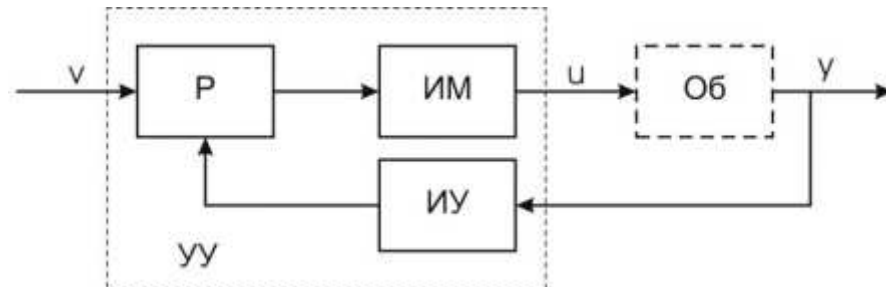


Рисунок 1 – Схема регулятора устройства управления

1.1 ТАУ

1.1.1 Управление от ЭВМ

В ТАУ управление выполняется различными средствами вычислительной техники в микроисполнении: однокристальные ЭВМ, процессоры, микроконтроллеры. Далее данные средства регулирования будем называть ЭВМ. Обобщенная схема управления от ЭВМ показана на рисунке 2. ОУ описывается данными состояния. Задачей управления является в каждый момент времени обеспечить требуемое состояние объекта управления.

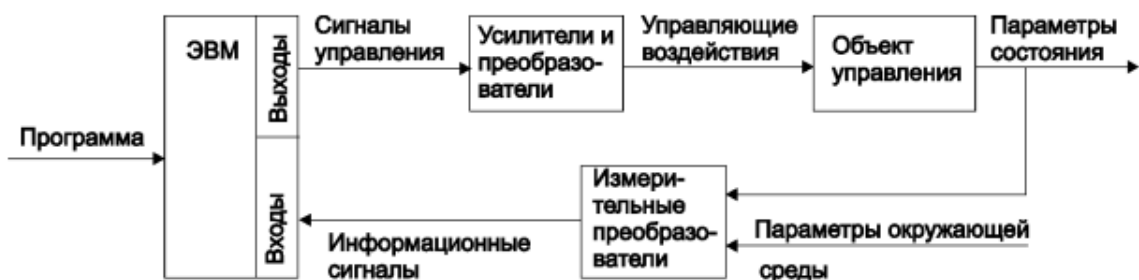


Рисунок 2 – Обобщенная схема управления от ЭВМ

Измерительные преобразователи используются для контроля состояния объекта, они преобразуют в информационные сигналы измеряемые параметры. Эти сигналы должны быть представлены в форме двоичного кода для автоматического ввода в ЭВМ. Аналоговые сигналы в двоичные коды преобразуются с помощью аналого-цифровых преобразователей. При решении задачи управления кроме параметров состояния могут учитываться параметры окружающей среды, отображающие условия функционирования объекта. Соответствующие измерительные преобразователи используются для контроля этих параметров. Параметры состояния объекта сравниваются с заданными значениями и определяется отклонение текущих параметров от заданных (рисунок 2). Значения требуемых управляющих воздействий определяются на основе вычисленных отклонений и информации об условиях функционирования, которые выводятся на выходы ЭВМ в виде числовых кодов. Программой определяется логика определения управления, что позволяет путем замены программы легко менять логику управления. Сигналы управления усиливаются, при необходимости преобразуются в аналоговые сигналы с помощью цифроаналоговых преобразователей и поступают на входы исполнительных механизмов объекта, меняя состояние последнего требуемым образом. ЦСУ содержат как непрерывные, так и дискретные сигналы, такие системы по принципу работы являются дискретными, что отличает их свойства от свойств непрерывных регуляторов. При проектировании системы управления необходимо обеспечить её работоспособность, т.е. система должна приводить объект

управления в заданное состояние во всех предусмотренных условиях его функционирования. Кроме того, система управления должна обеспечивать требуемое состояние объекта с заданной точностью и иметь требуемое быстродействие, т.е. система управления должна обладать необходимым качеством[8].

1.1.2 Автоматическое регулирование

Задачи управления различны и их сложность определяет сложность применяемых систем управления. Управление может осуществляться вручную и автоматически. Автоматическое управление – осуществление совокупности воздействий на объект управления, выбранных на основании определенной информации из множества возможных и направленных на поддержание или улучшение функционирования управляемого объекта в соответствии с задачей управления. Автоматическое регулирование является частной задачей автоматического управления. Автоматическое регулирование – поддержание постоянной или изменение по заданному закону некоторой величины, характеризующей процесс, осуществляемое при помощи измерения состояния объекта или действующих на него возмущений и воздействия на объект. Задачу разработки САР приходится часто решать при проектировании механических устройств. Задача эта имеет достаточную сложность, и её решение требует использования методов ТАУ.

Можно рассматривать как базовый метод управления автоматическое регулирование с целью обеспечения заданного значения управляемой величины на выходе объекта управления. САР являются основой САУ. Управление в САР осуществляется некоторым объектом с целью получения необходимого результата (рисунок 3), этот результат заключается в обеспечении заданного состояния объекта. Состояние объекта характеризуется значением его выходной (управляемой) величины.

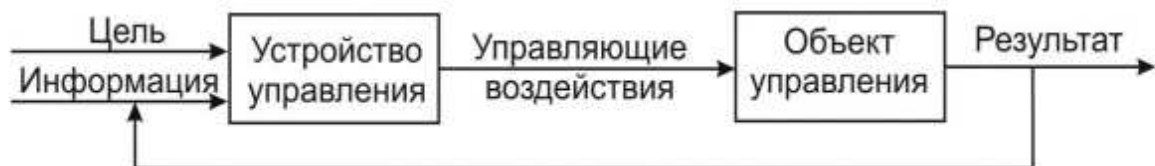


Рисунок 3 – Система автоматического регулирования

Управляющие воздействия подаются на вход управления для изменения состояния объекта. Устройством управления на основе цели управления формируются управляющие воздействия. ТАУ включает аналитические модели и методы, позволяющие реальные автоматические системы описать и исследовать с целью определения их поведения в автоматическом режиме работы и создания систем с требуемыми свойствами. ТАУ решает задачи аналитического описания автоматических систем, их анализа и синтеза[8].

1.1.3 Обобщенная САУ

ОУ описывается выходными управляемыми величинами y_1, y_2, \dots, y_n . Истинное состояние объекта управления определяют показания этих величин в определенный момент времени. Кроме того, объект имеет входы управления, на которые подаются управляющие величины u_1, u_2, \dots, u_n , умышленно изменяющие состояние объекта. Обеспечение нужного состояния ОУ в данный момент времени и есть цель управления, то есть заложенные значения управляемых величин объекта. УУ, как следует из заложенного состояния ОУ, определяемого заложенными величинами v_1, v_2, \dots, v_n , и действующего его состояния, определяемого выходными величинами y_1, y_2, \dots, y_n , вырабатывает управляющее воздействия u_1, u_2, \dots, u_n на объект управления так, чтобы привести его в данное состояние $y_1 = v_1, y_2 = v_2, y_n = v_n$ и при всем этом совокупность возмущений на объект оказывают влияние, выводящих его из данного состояния. Все возмущения можно поделить на

контролируемые $g_1, g_2 \dots g_m$ и неконтролируемые $f_1, f_2 \dots f_k$ (f - помехи, g - нагрузки). Возмущения нарушают работу объекта и приводят к ненужным изменениям $y_1, y_2 \dots y_n$. Задачей УУ является обеспечение в этих критериях требуемых значений $y_1, y_2 \dots y_n$, поэтому, САУ включает в себя УУ и ОУ. Вместо САУ, при автоматическом управлении, рассматривается САР. Часто ОУ определяется одной выходной величиной и одним входом управления. В данном случае говорят об одном контуре управления (регулирования) и объект относят к обычным объектам. Объект управления может иметь много выходных величин и входов управления, если каждый вход управления повлияет лишь на одну выходную единицу и изменение хоть какой выходной единицы ОУ не влияет на состояние иных выходов объекта, то каждый контур управления можно рассматривать обособленно и рассматривать объект как объединения подходящего числа обычных объектов. Но для ряда объектов с несколькими выходами изменение одной выходной единицы приводит к изменению и прочих выходных единиц. Выходные единицы и входы управления у такого объекта взаимосвязаны, объект тогда называют многосвязным объектом. Задача управления многосвязными объектами значительно труднее. При управлении обычным объектом САУ либо самостоятельный контур управления можно представить в виде обобщенной обычной структуры (рисунок 4).

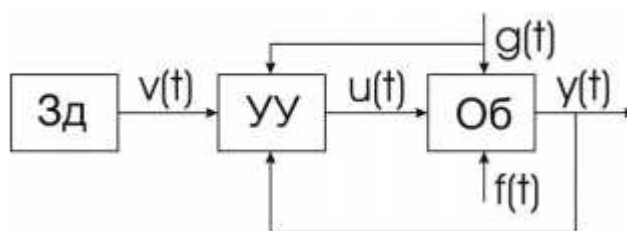


Рисунок 4 – Обобщенная упрощенная структура САР

Управляющее воздействие - воздействие $u(t)$ на объект управления Об, выдающее УУ. Возмущения - воздействия $g(t), f(t)$, не зависящие от УУ и вызывающее нежелательное изменение состояния объекта. Возмущения

делятся на управляемыми $g(t)$ (нагрузки) и неуправляемыми $f(t)$ (помехи). Управляемая или регулируемая величина — это выходная единица объекта $y(t)$, по которой ведется управление. Данное значение регулируемой величины определяется воздействием $v(t)$, поступающим на вход УУ. Задатчик Зд сформировывает задающее воздействие. Это воздействие называют задающим воздействием или уставкой. Устройство автоматического управления в возникнувшем случае автоматического регулирования называют автоматическим регулятором. В состав автоматического регулятора УУ входят: измерительное устройство ИУ, логическое (усилительно-преобразующее) устройство Р и исправный механизм ИМ (рисунок 5).

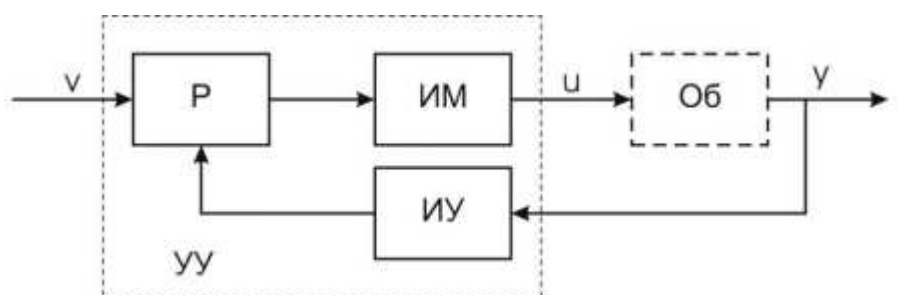


Рисунок 5 – Схема регулятора устройства управления

Логическое устройство Р сравнивает действующее значение управляемой величины $y(t)$ с данным значением $v(t)$ этой величины и по определённому алгоритму сформировывает сигнал, необходимый для устранения расхождения между этими величинами. Для контроля состояния объекта управления используется измерительное устройство. С помощью данного устройства измеряется управляемая величина и на платформе управления формируются нужные данные о состоянии объекта управления. Сигнал управления преобразуется в соответствующее физическое управляющее воздействие на объект управления при помощи исполнительного механизма. Законом регулирования называют метод формирования управляющего воздействия[8].

1.2 Программирование

1.2.1 C++

C++ — компилируемый, статически типизированный язык программирования общего назначения.

В данном языке поддерживаются такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. Язык имеет широкую стандартную библиотеку, в которую включены распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения и другие возможности. C++ сочетает свойства высокоуровневых и низкоуровневых языков. В данном языке программирования наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования.

Обширно используется C++ для разработки ПО, являясь одним из самых популярных языков программирования. Область применения языка включает в себя создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр. Существует множество исполнений языка C++ бесплатных и коммерческих для различных платформ. Например, на платформе x86 это GCC, Visual C++, Intel C++ Compiler, Embarcadero (Borland) C++ Builder и другие. C++ оказал огромное влияние на другие языки программирования.

Синтаксис C++ наследован от языка C. Одним из принципов разработки было сохранение совместимости с C. Тем не менее C++ не является надмножеством C. Много программных кодов, которые могут успешно транслироваться компиляторами C и C++, довольно велико, но не включает все возможные программы на C[17].

1.2.2 Java

Программы на Java транслируются в байт-код Java, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

Преимуществом такого метода выполнения программ является абсолютная независимость байт-кода от ОС и оборудования, что позволяет выполнять приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Гибкая система безопасности — это еще одна важная особенность технологии Java, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы, вызывают немедленное прерывание.

Снижение производительности очень часто относят к минусам концепции виртуальной машины. Некоторые модернизации несколько увеличили скорость работы программ на Java: применение технологии трансляции байт-кода в машинный код во время работы программы с возможностью сохранения версий класса в машинном коде, обширное использование платформенно-ориентированного кода в стандартных библиотеках, аппаратные средства, обеспечивающие ускоренную обработку байт-кода.

По данным сайта shootout.alioth.debian.org, для семи разных задач время выполнения на Java в полтора-два раза больше, чем для C/C++, в некоторых случаях Java быстрее, а в некоторых случаях в 7 раз медленнее. С другой стороны, потребление памяти Java для большинства из них было в 10—30 раз больше, чем программой на C/C++. Исследование, проведенное компанией Google, согласно которому отмечается более низкая производительность и увеличенное потребление памяти в текстовых примерах на Java в сравнении с программами на C++.

Идеи, заложенные в концепцию и различные реализации среды виртуальной машины, воодушевили множество энтузиастов на расширение перечня языков, которые могли бы быть использованы для создания программ, исполняемых на виртуальной машине. Эти идеи нашли выражение в спецификации общезыковой инфраструктуры CLI, заложенной в основу платформы .NET компанией Microsoft[18].

1.2.3 Python

Язык использует динамическую типизацию вместе с подсчётом ссылок и циклический сборщик мусора для менеджмента памяти. Также есть динамические разрешения имен, которые связывают имена методов и переменных во время выполнения программы.

Python предлагает поддержку функционального программирования в традициях Лиспа. Так, в Python есть функции `filter`, `map` и `reduce`; также из Лиспа были заимствованы понятия характеристик списков, ассоциативных массивов, множеств и генераторов списков. Стандартная библиотека содержит два модуля: `itertools` и `functools`, реализующие инструменты, заимствованные из Haskell и Standard ML.

Разработчики языка Python придерживаются определённой философии программирования, называемой «The Zen of Python». Её текст выдаётся интерпретатором Python по команде `import this`. Автором этой философии считается Тим Петерс.

Вместо того, чтобы встроить в ядро Python всю функциональность языка, он был разработан таким образом, чтобы быть легко расширяемым. Это сделало язык популярным средством прибавления программируемых интерфейсов к имеющимся прибавлениям. Видение Гвидо Ван Россума маленького ядра с великой стандартной библиотекой и просто расширяемым интерпретатором происходило из отрицательного опыта разработки языка ABC, который придерживался обратного подхода.

Python стремится к более обычному, меньшему синтаксису и грамматике, предоставляя разработчикам выбор в их методологии программирования. В небольшом отличии от лозунга Perl есть несколько методов сделать это, Python придерживается философии должен существовать один очевидный метод сделать это. Алекс Мартелли, член Python Software Foundation, и создатель книг по Python пишет, что «Описывать что-то как «умное», не считается комплиментом в культуре Python»).

Разработчики Python устремляются избежать ненужной оптимизации и опровергают патчи к некритическим долям эталонной реализации CPython, которые имели возможность бы предложить небольшое увеличение скорости за счёт понятности кода. Однако есть методы роста производительности. Если, предположим, в коде имеются тесные места, связанные с выполнением ресурсоёмких операций на центральном микропроцессоре, но не имеющие связь с применением операций ввода-вывода, то повысить производительность можно с помощью трансляции программы при помощи Cython в язык C и последующей компиляции. Взыскательные к вычислительным ресурсам части программы также конечно возможно переписывать на язык C и подключать как отдельные библиотеки с привязками к Python.

Основная цель разработчиков Python делать его забавным для использования. Также это отражено в иногда игривом подходе к обучающим программам и справочным материалам, таким как примеры программ из документаций, которые используют названия `spam` и `eggs` вместо использующихся в документации множества других языков `foo` и `bar`.

Python портирован и работает практически на абсолютно всех узнаваемых платформах от КПК до мейнфреймов. Существуют порты под Microsoft Windows, фактически под все варианты UNIX, Android, Plan 9, Mac OS и macOS, iPhone OS (iOS) 2.0 и выше, iPadOS, Palm OS, OS/2, Amiga, HaikuOS, AS/400, OS/390, Windows Mobile и Симбиан.

По мере устаревания платформы её поддержка в основной ветви языка прекращается. К примеру, с версии 2.6 прекращена поддержка Windows 95, Windows 98 и Windows ME. В версии 3.5 закончила поддерживаться Windows XP В версии 3.9 перестала поддерживаться Windows Vista и Windows 7.

При всем этом, в небольшом отличии от многих портируемых систем, для абсолютно всех главных платформ Python имеет поддержку отличительных для данной платформы технологий (к примеру, Microsoft COM/DCOM). Плюс к этому, существует особая версия Python для виртуальной машины Java Jython, что дает возможность интерпретатору запускаться на любой системе, поддерживающей Java, при всем этом классы Java могут конкретно употребляться из Python причем даже быть написанными на Python. Также несколько проектов дают обеспечение интеграцию с платформой Microsoft.NET, главные из которых IronPython и Python.Net[19].

1.2.4 Arduino

Программирование ведется полностью через свою бесплатную программную оболочку Arduino IDE. В этой оболочке имеется текстовый редактор, менеджер проектов, препроцессор, компилятор и приборы для загрузки программ в микроконтроллер. Оболочка создана на языке Java на базе проекта Processing, работает под Windows, Mac OS X и системы Linux. Употребляется комплект библиотек Arduino.

Язык программирования Arduino называется Arduino C и представляет язык C++ с фреймворком Wiring, он имеет отличия в написании кода, который компилируется и собирается при помощи avr-gcc, с методикой, облегчающей создание работающей программы, есть набор библиотек, включающий в себя функции и объекты. При компиляции программы IDE создает временный файл с расширением *.cpp.

Программы, написанные программистом на Arduino, называются скетчами и сохраняются в файлах с расширением *.ino. Эти файлы перед компиляцией обрабатываются препроцессором Arduino. Файлы C++ имеют возможность создания и подключения к проекту.

Программист должен написать две неотъемлемые для Arduino функции `setup()` и `loop()`. 1-ая вызывается однократно при старте, 2-ая выполняется в бесконечном цикле.

В текст программист не должен вставлять заголовочные файлы применяемых стандартных библиотек. Эти заголовочные файлы добавит препроцессор Arduino в соответствии с конфигурацией проекта, но пользовательские библиотеки необходимо указывать.

Библиотеки на стандартном C++ добавляются в специальную рабочую папку в рабочем каталоге IDE. И при всем этом название библиотеки добавляется в перечень библиотек в меню IDE. Программист отмечает нужные библиотеки, и они вносятся в перечень компиляции[20].

2 Примеры датчиков

2.1 САРД G259

2.1.1 Описание САРД G259

В САРД встроен радиолокационный датчик, для расчета дистанции, работающий в спектре миллиметровых волн. Этот датчик создает замеры расстояний сразу до нескольких объектов, находящихся в зоне его охвата, и после этого рассчитывает условную скорость объектов, через которые проходит продольная ось ТС. Угол отклонения (азимут) объекта от оси зоны охвата датчика рассчитывается по данным замеров, изображение датчика представлено на рисунке 6.



Рисунок 6 – Датчик САРД G259

САРД использует электрические волны, распространяющиеся со скоростью света c , каждой частоте f соответствует определенная длина волны. Передатчик САРД работает на частоте $f=76,5$ ГГц, которой

соответствует длина волны $\lambda = 3,92$ мм. Волны в частотном спектре от 30 до 150 ГГц называются миллиметровыми (рисунок 7).

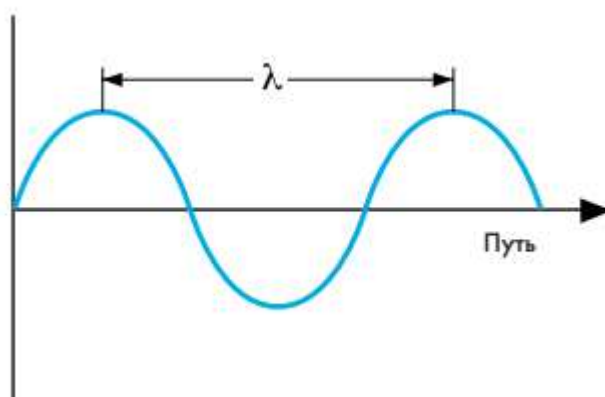


Рисунок 7 – Длина волны датчика

В корпусе САРД расположено вычислительное устройство, позволяющее расчетным методом найти:

- упреждающее положение полосы следования;
- объект слежения;
- данные регулирования дистанции и скорости ТС;
- данные для передачи на блоки управления двигателем, усилителем тормозного привода и композицией устройств;
- диагностические характеристики[3].

2.1.2 Усилитель тормозного привода с электронным управлением

Для конфигурации расстояния до передвигающегося впереди транспортного средства путем торможения ТС с САРД предусмотрен усилитель тормоза с электрическим управлением. В усилитель с расположенными один за другим вакуумными камерами встроены: пропорционально работающий электромагнит, датчик положения диафрагмы усилителя и выключатель САРД. Вызываемое им торможение обязано быть плавным и не ухудшающим комфортабельность ТС. Якорь электромагнита движется пропорционально силе проходящего через обмотку тока.

Бесступенчатый потенциометр употребляется в качестве датчика положения[3].

2.1.3 Выключатель САРД

Для определения состояния системы предусмотрен выключатель САРД, точнее, делается ли торможение водителем либо оно происходит в ходе срабатывания электромагнита. Так как данный выключатель является элементом безопасности ТС, он обустроен 2 парами контактов, по расположению которых определяется исходное или рабочее положение усилителя.

Исходное положение усилителя характеризуется наличием вакуума почти во всех его полостях и обесточенным пропорционально действующим электромагнитом.

Усилителем тормозного привода управляет Блок управления САРД посредством клапана, уплотняющая кромка которого ведет взаимодействие с кольцевой прокладкой его седла. Давление в атмосферной полости усилителя зависит от места положения этого клапана[3].

2.1.4 Давление в тормозной магистрали

Блок управления САРД принуждает возрасти давление в тормозной магистрали, подключая работающий электромагнит к источнику тока. Через раскрывающийся впускной клапан наружный воздух поступает в атмосферную полость усилителя, при этом сокращается зазор между магнитопроводом и якорем электромагнита, в итоге его диафрагма перемещается, сжимая возвратную пружину. При всем этом тормозное усилие может достичь 30 процентов от его максимального значения.

При обесточенной обмотке якорь электромагнита упирается уплотняющей кромкой в прокладку седла и отжимает ее от уплотняющей

кромки выпускного клапан, в итоге воздух из атмосферной полости перемещается в вакуумную полость и далее отсасывается во впускную систему двигателя. Пружина диафрагмы при всем этом разжимается.

Сниженный ток обмотки электромагнита употребляется для поддержания давления в тормозной магистрали. Пружина якоря отжимает его от магнитопровода и закрывает впускной клапан. Положение диафрагмы усилителя и при всем этом может зависеть от количества пониженного давления воздуха в рабочей полости[3].

2.1.5 Обмен данными через шину CAN

Датчик САРД связан с шиной данных CAN через реле, встроенное в блок управления усилителем тормозного привода. В систему обмена данными с датчиком САРД включены следующие приборы (рисунок 8 – 13) [3]:



Рисунок 8 - Блок управления усилителем тормозного привода



Рисунок 9 - Блок управления двигателем

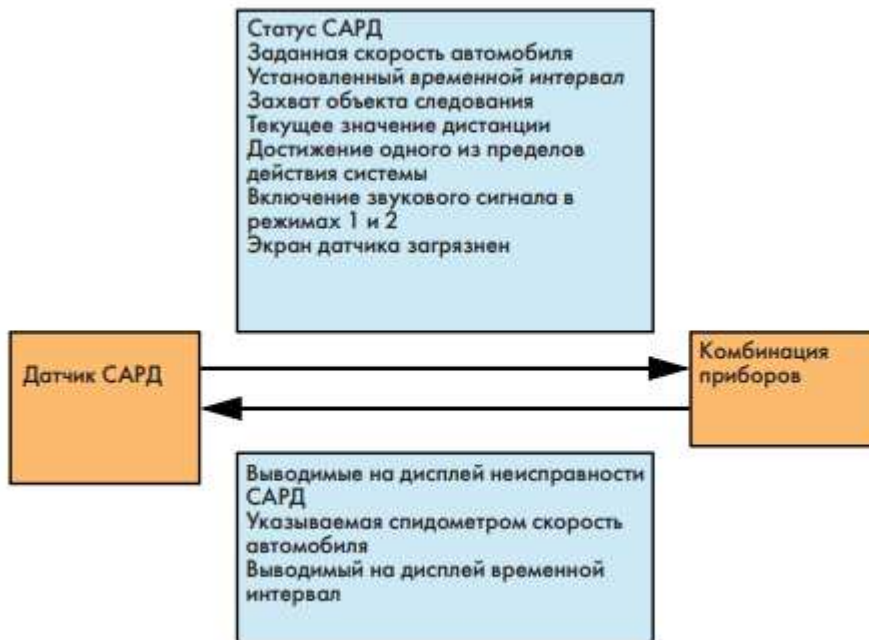


Рисунок 10 - Комбинация приборов

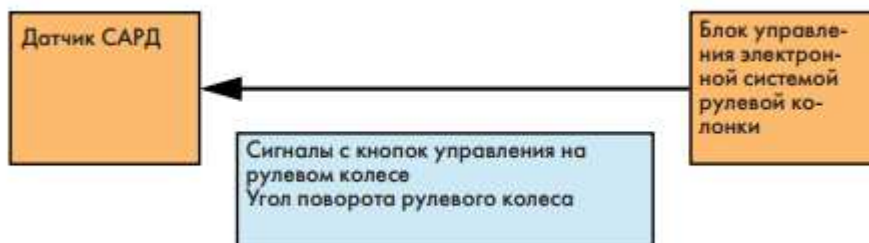


Рисунок 11 - Блок управления электронной системой рулевой колонки



Рисунок 12 - Блок управления автоматической коробкой передач

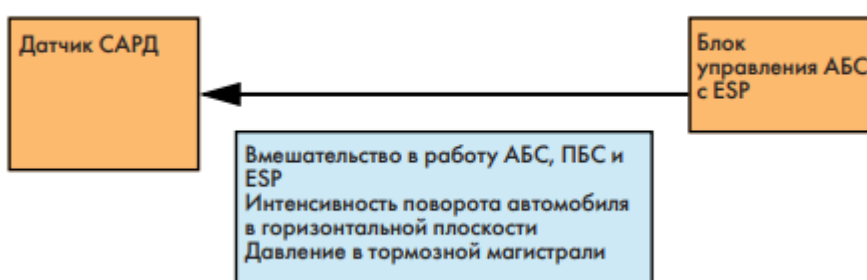


Рисунок 13 - Блок управления ABS с ESP

2.1.6 Функциональные возможности системы

Если, предположим, в области охвата датчика САРД транспортных средств нет, поддерживается данная скорость.

Если ТС с САРД при движении по собственной полосе настигает движущееся медленнее его ТС, система САРД понижает скорость посредством убавления вращающего момента мотора и, по мере надобности, методом осторожного притормаживания. И при всем этом обеспечивается все постоянство задаваемого водителем времени, которое нужно для преодоления расстояния до передвигающегося впереди средства передвижения (рисунок 14).

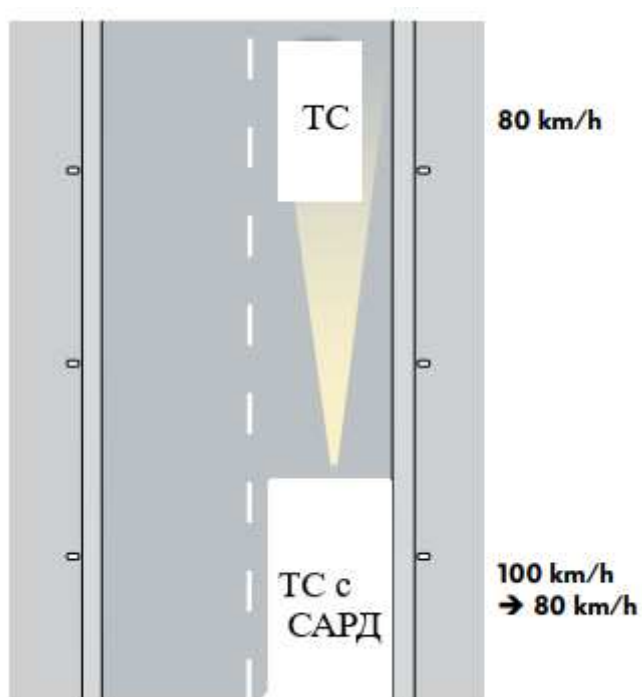


Рисунок 14 – Автоматическое торможение

По такому же принципу скорость ТС с САРД сравнивается со скоростью более медленного средства передвижения, которое оказалось перед ТС в итоге перестройки из одной полосы движения в другую.

Если движущееся впереди ТС наращивает свою скорость или меняет полосу, ТС с САРД опять разгоняется до данной первоначально скорости (рисунок 15) [3].

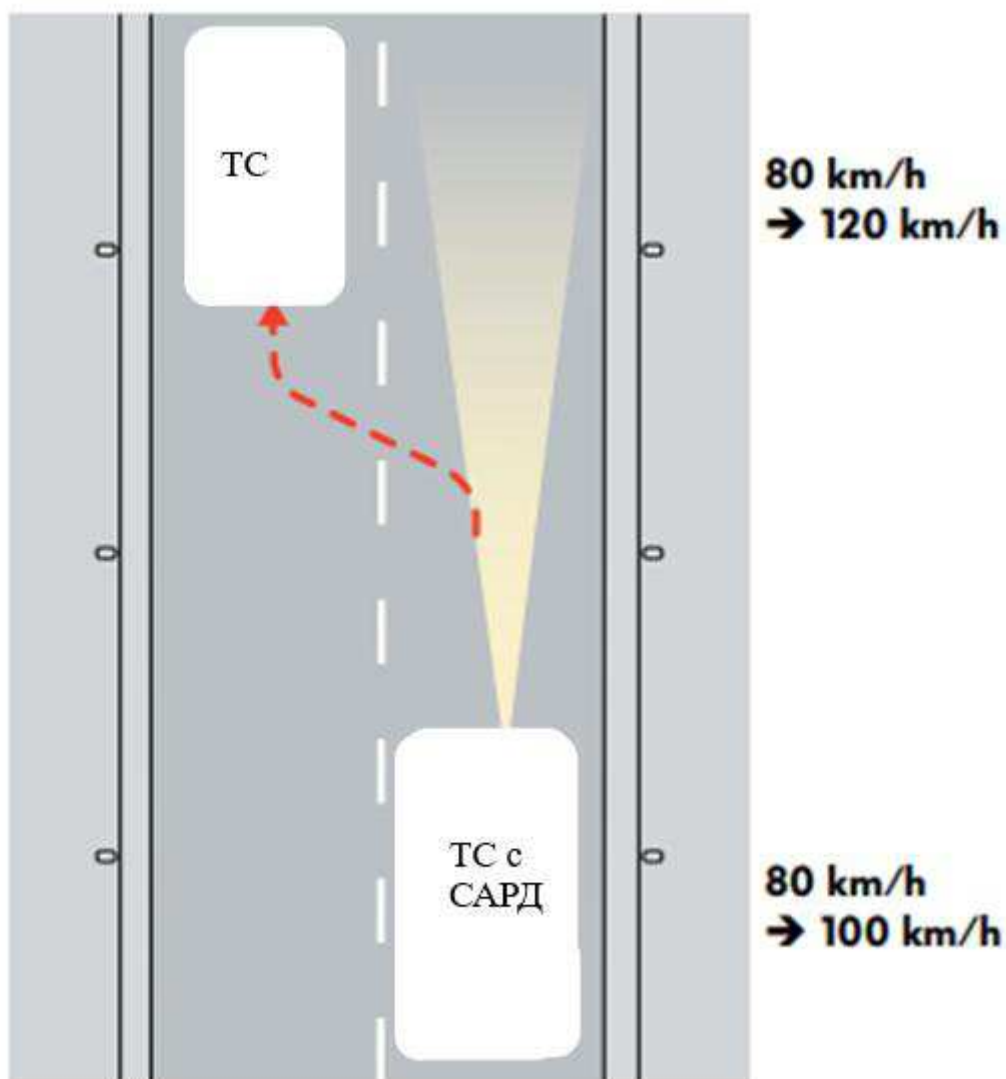


Рисунок 15 – Автоматическое ускорение

2.2 Лазерный датчик

2.2.1 Lidar

Лидар - технология принятия и обработки информации об дальних объектах строительства с помощью активных оптических систем, использующих явления отражения света и его рассеяния в прозрачных и полупрозрачных средах. Двумерную или трёхмерную картину окружающего места формируют сканирующие лидары в системах машинного зрения.

Определяют дальность до непрозрачных отражающих целей и рассматривают характеристики прозрачной среды атмосферные лидары.

Разнообразием атмосферных лидаров являются доплеровские лидары, определяющие направление и скорость передвижения воздушных потоков в разных слоях атмосферы. Принцип работы лидара представлен на рисунке 16.

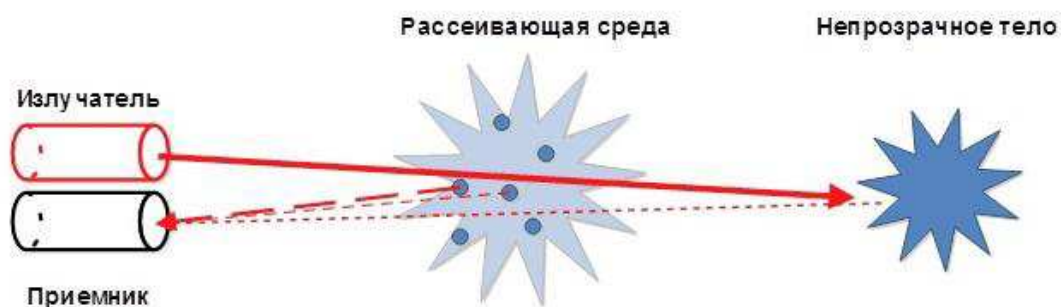


Рисунок 16 – Принцип действия лидара

Работа лидара не слишком сильно отличается от радара: направленный луч источника излучения отражается от точек, возвращается к источнику и улавливается высокочувствительным приёмником (в случае лидара светочувствительным полупроводниковым устройством); время отклика прямо пропорционально расстоянию до цели.

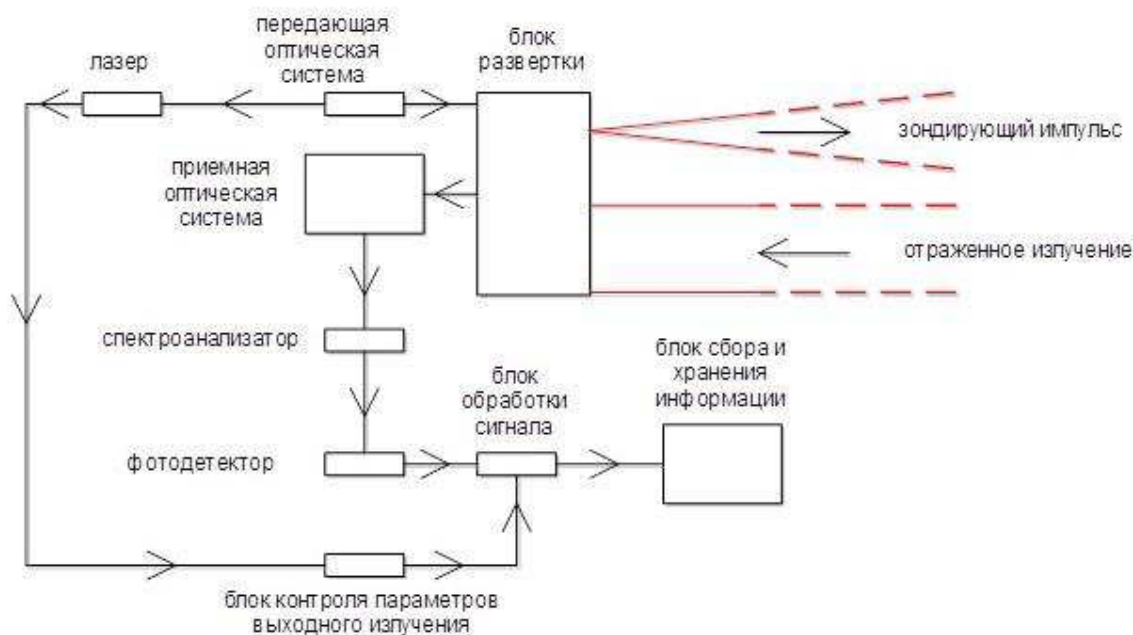


Рисунок 17 – Схема простейшей лидарной системы

Принцип работы лидара прост, объект (поверхность) освещается коротким световым импульсом, и замеряется время, через которое сигнал возвратится к источнику. Свет распространяется очень быстро - 3108 м/с, но он возвращается с некой задержкой, зависящей от расстояния до объекта. Лидар запускает быстрые недлинные импульсы лазерного излучения на объект (поверхность) с частотой до 150000 импульсов в секунду. Датчик на приборе замеряет время, нужное для возврата импульса. Свет движется с неизменной скоростью, поэтому лидар может высчитать расстояние между ним и целью точно. Есть 2 главные категории импульсных лидаров: микроимпульсные и высокоэнергетические системы.

— Микроимпульсные лидары работают на более высокой компьютерной технике с большими вычислительными способностями. Эти лазеры наименьшей мощности и классифицируются как "неопасные для глаз", что дает возможность использовать их без особенных мер безопасности.

— Лидары с большой энергией импульса используются для исследования атмосферы, где они часто и используются для замеров различных характеристик атмосферы, таких как высота, наслоение и плотность облаков, свойства частиц облака, температуру, давление, ветер, влажность и концентрацию газов в атмосфере.

В малом отличии от радиоволн, отлично отражающихся только от крупных металлических целей, световые волны подвержены рассеиванию во всех средах, в том числе и в воздухе, поэтому возможно определять не только расстояние до непрозрачных дискретных целей, но и фиксировать интенсивность рассеивания света в прозрачных средах. Отражённый сигнал проходит через ту же рассеивающую среду, что и луч от источника, подвергается повторному рассеиванию, поэтому восстановление реальных

характеристик оптической среды достаточно непростая задача, решаемая как аналитическими, так и эвристическими методами[6].

2.2.2 Излучатель

В конструкциях излучателем служит лазер, формирующий недлинные импульсы света высокой мощности. Периодичность следования импульсов или модулирующая частота выбираются так, чтобы пауза между 2 поочередными импульсами была не меньше, чем время отклика от обнаружимых целей (которые могут физически находиться дальше, чем расчётный радиус работы устройства). Выбор длины волны может зависеть от функции лазера и требований к безопасности и скрытности прибора; более довольно частенько используются Nd:YAG-лазеры и последующие длины волн (в нанометрах):

- 1550 нм — инфракрасное излучение, невидимое ни глазу человека, ни типичным приборам ночного видения. Глаз не способен сфокусировать эти волны на поверхности сетчатки, поэтому травматический порог для волны 1550 существенно больше, чем для более коротких волн;
- 1064 нм — ближнее инфракрасное излучение неодимовых и иттербиевых лазеров, невидимое глазу, но обнаружимое приборами ночного видения;
- 532 нм — зелёное излучение неодимового лазера, эффективно «пробивающее» массы воды;
- 355 нм — ближнее ультрафиолетовое излучение.

Также конечно возможно внедрение вместо кратких импульсов постоянной амплитудной модуляции излучения переменным напряжением[6].

2.2.3 Сканирующая оптика

Простые атмосферные лидарные системы не имеют средств наведения и ориентированы вертикально в зенит.

Для сканирования горизонта в некой плоскости используются обычные сканирующие головки. В них неподвижные излучатель и приёмник также ориентированы в зенит, под углом в 45 к горизонту и полосы излучения установлено зеркало, вращающееся вокруг оси излучения. В авиационных установках, в каком месте нужно исследовать полосу, перпендикулярную направлению полёта самолёта-носителя, ось излучения горизонтальна. Для осуществления синхронизации мотора, крутящего зеркало, и средств обработки принимаемого сигнала употребляются четкие датчики положения ротора, но также неподвижные реперные опасности, наносимые на прозрачный кожух сканирующей головки.

Сканирование в 2-ух плоскостях прибавляет к этой схеме механизм, поворачивающий зеркало на фиксированный угол с каждым оборотом головки так формируется цилиндрическая развёртка находящегося вокруг мира. Присутствие достаточной вычислительной мощности можно использовать жёстко закреплённое зеркало и пучок расходящихся лучей в таковой конструкции один кадр формируется за один оборот головки.

Важную роль играет динамический спектр приёмного тракта. Чтоб избежать перегрузки приёмника насыщенной засветкой от рассеивания в ближней зоне, в системах далекого радиуса деяния употребляют высокоскоростные механические затворы, физически блокирующие приёмный оптический канал. В устройствах ближнего радиуса со временем отклика менее микросекунды такой способности нет[3].

3 Система с ультразвуковым датчиком

3.1 Описание

Ультразвуковой датчик расстояния в большинстве случаев употребляется для обнаружения объектов и измерения расстояния до них. Принцип работы устройства таков: устройство выдает звуковые колебания определенной частоты, при скорой встрече с твердой поверхностью выпущенные звуковые волны возвращаются назад в датчик. После этого микроконтроллер рассчитывает расстояние до объекта по определенной формуле[7].

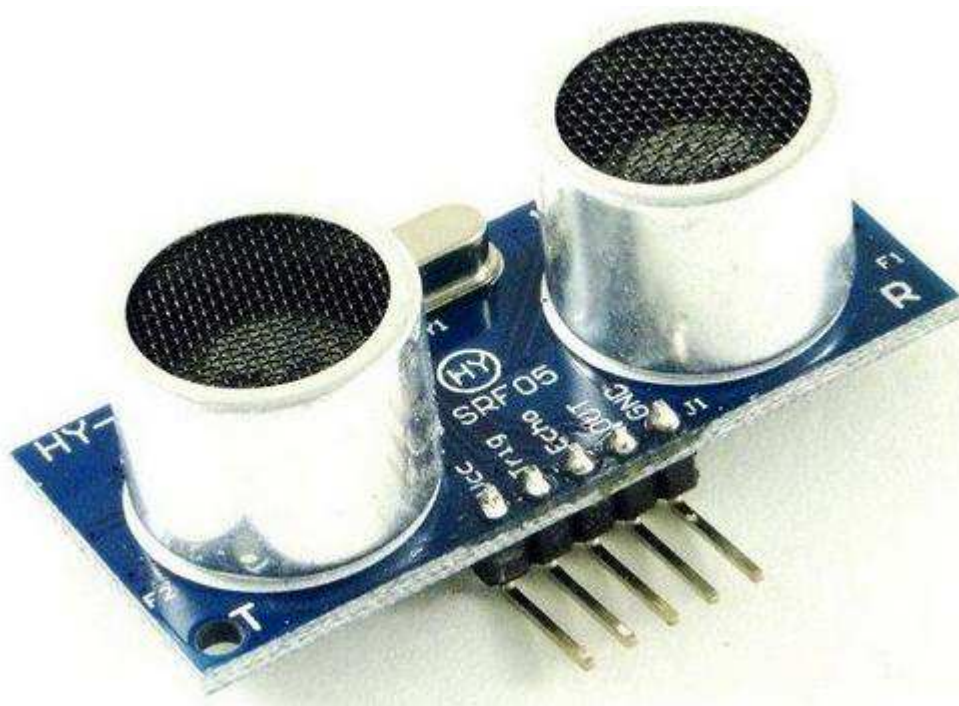


Рисунок 18 – Ультразвуковой датчик

При высоких скоростях ТС нужен увеличенный путь торможения, то есть его торможение обязано быть начато при большей, чем указана выше, дистанции до передвигающегося впереди транспортного средства. Если ТС со встроенной системой находится в стадии пассивного торможения с условно большей скорости, при понижении ее до нижнего граничного

значения на экран композиции устройств выводится требование, приступить водителю к активному торможению.

Поступающие с колесных датчиков сигналы частоты вращения передаются на вход блока управления Abs с ESP и дальше на датчик системы. Эти сигналы необходимы для правильного распознаванию полосы движения.

Управление системы производится главным образом посредством кнопок на функциональном мониторе и частично средством тормоза, как это не исключается при управлении системой регулирования скорости ТС. Кнопки на экране соединены с блоком управления электрической системы управляющий оси, которая передает сигналы на комбинацию устройств.

Чтобы водитель мог в любой момент времени узнать о состоянии системы, на экран комбинации приборов показываются последующие данные, сопровождаемые звуковыми сигналами:

- статус системы;
- назначенные водителем параметры;
- предупреждающие указания.

Система управляется в целом посредством кнопок, расположенных на дисплее, но она получает команды также от тормоза. После каждого запуска мотора система находится в неработающем состоянии; она приводится в состояние активности нажимом кнопки ON/ OFF.

Нажимом кнопки RES (возобновление) система включается с переходом на ранее заданную скорость. При каждом повторном нажмении кнопки RES записанная скорость увеличивается на 1 км/ч прямо до ее максимального значения значения. Нажимая кнопки GRA+ и GRA-, можно также поднять либо уменьшить заданную скорость с шагом 10 км/ч. Водитель с удовлетворением принимает автоматическое изменение дистанции до последующего далее ТС в большой зависимости от скорости движения. С ростом скорости дистанция должна возрастать. И при всем этом

время преодоления дистанции ТС со встроенной системой остается неизменным, вот поэтому этот период является мерой изменяемой со скоростью дистанции. Оно именуется также временным интервалом[3].

На рисунке 19 приведены различные датчики и схемы их подключения.

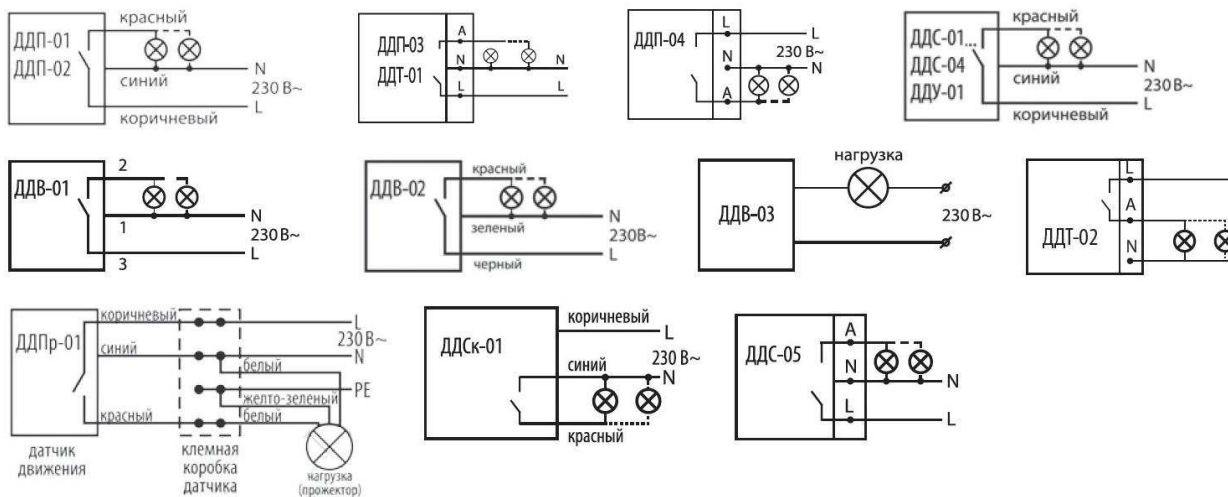


Рисунок 19 – Схемы подключения различных датчиков

Схема датчика с микроконтроллером, подающим данные в блок управления, представлена на рисунке 20.

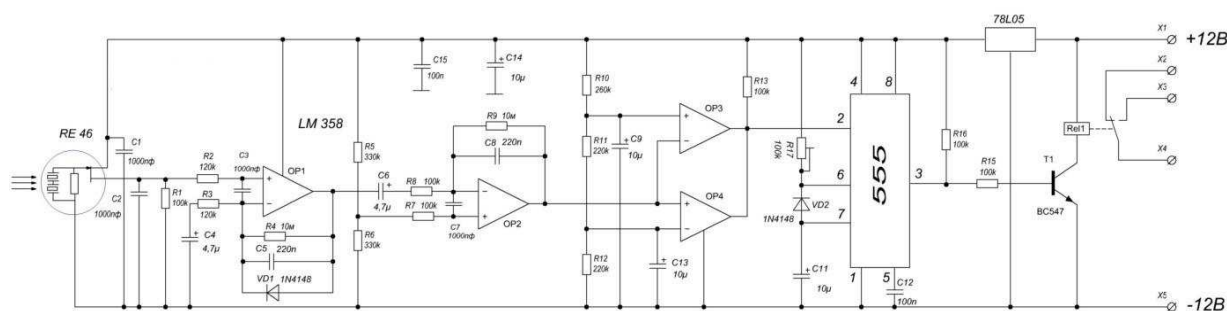


Рисунок 20 – Схема датчика с микроконтроллером

3.2 Arduino Uno

В базе платы находится процессор ATmega 328. Не считая него на плате размещен модуль USB для связи с ПК и прошивки. Этот модуль

называется USB-TTL преобразователь. На фирменных платах Arduino Uno для этих целей находится дополнительный микроконтроллер ATmega16U2.

Характерная особенность этого чипа содержится в аппаратной поддержке USB, что дает возможность организовывать связь без дополнительных преобразователей, когда как ATmega328 не поддерживает таковой функции, потому 16u2 находится в роли преобразователя данных из USB в поочередный порт для МК AVR. В него залита программа для исполнения этой задачи.

Чтобы включить плату, необходимо на неё дать питание либо от USB, либо прямо от ПК, либо от другого источника питания от 7 до 15 Вольт. На плате установлен линейный стабилизатор, типа L7805, либо же LDO, он нужен затем, чтоб на микроконтроллер подавалось стабилизированное напряжение 5 В.

И при всем этом выбирается наружный источник питания, но не USB-порт. Наружное питание подключается к выводу с пометкой Vin в разделе Power на плате.

Микроконтроллер имеет четырнадцать цифровых пинов, им предоставляется возможность быть использованными как вход либо выход. Из их 6 могут выдавать ШИМ-сигнал. Они необходимы для регулировки мощности в нагрузке и прочих функций[5].

Подключение к Arduino для настройки датчика показано на рисунке 21.

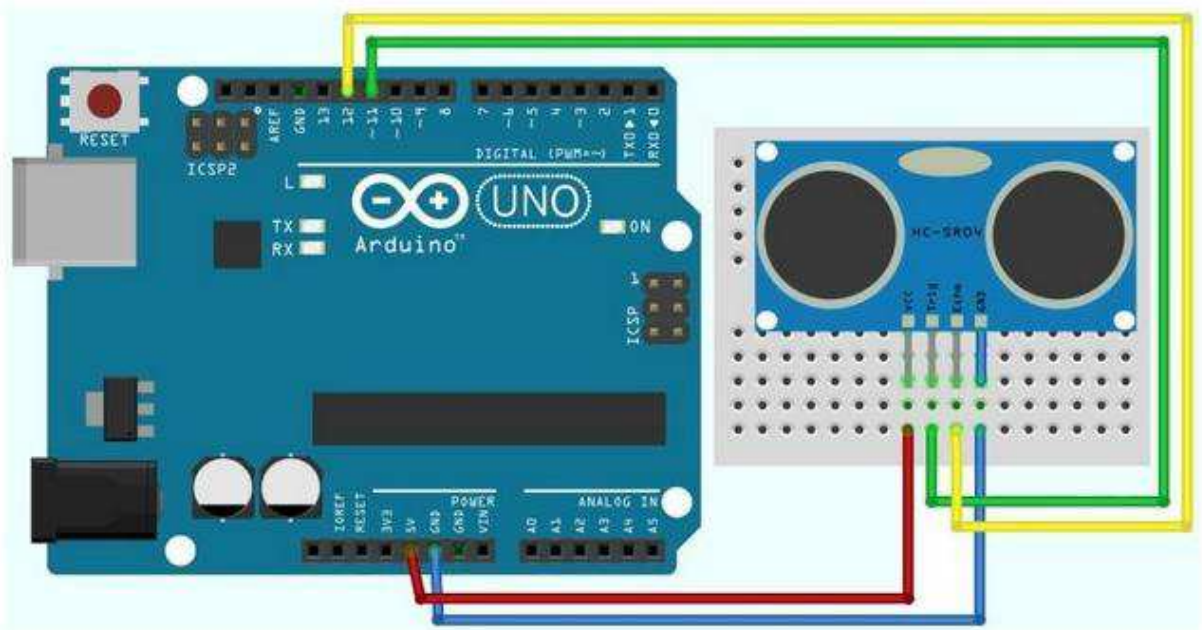


Рисунок 21 – Подключение датчика к Arduino

ЗАКЛЮЧЕНИЕ

В выпускной квалификационной работе были исследованы основы предотвращения столкновений электросамоката, его системы управления, а также инструментария для осуществления данного процесса. Исходя из полученных данных, можно утверждать, что большинство систем предотвращения столкновений для электросамокатов были разработаны благодаря богатому опыту, накопленному при производстве автомобилей, мотоциклов, мопедов и многих других транспортных средств.

Все цели и задачи работы достигнуты. Были изучены и разработаны датчики предотвращения столкновений электросамоката.

При выполнении данной работы был произведен обзор предметной области, задания на выпускную квалификационную работу, изучены существующие аналоги и сформированы требования, предъявляемые к датчикам регулировки скорости.

СПИСОК СОКРАЩЕНИЙ

- САР – Система автоматического регулирования;
- САУ – Система автоматического управления;
- ТАУ – Теория автоматического регулирования;
- ОР – Объект регулирования;
- ОУ – Объект управления;
- ЭВМ – Электронная вычислительная машина;
- ЦСУ – Цифровые системы управления;
- УУ – Устройство управления;
- ТС – Транспортное средство;
- САРД – Система автоматического регулирования дистанции.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Что такое электросамокат [Электронный ресурс]. – Режим доступа: <https://e-samokat.com/chto-takoe-elektrosamokat>.
- 2 Все про контроллеры электросамоката | схема, проверка, ремонт [Электронный ресурс]. – Режим доступа: <https://www.voltbikes.ru/blog/about-segways/vse-pro-kontrollery-jelektrosamokata/>.
- 3 Система автоматического регулирования дистанции (САРД) [Электронный ресурс]. – Режим доступа: <http://almarka.ru/wp-content/uploads/2019/08/avtomobil-phaeton-sistema-avtomaticheskogo-regulirovaniya-distancii-sard-276.pdf>
- 4 Система автоматического торможения для предотвращения столкновений: суть системы и принципы работы [Электронный ресурс]. – Режим доступа: <https://naavtotrasse.ru/encziklopediya/sistema-avtomaticheskogo-tormozheniya-dlya-predotvrashheniya-stolknovenij.html>
- 5 Arduino Uno R3 на базе процессора ATmega328 [Электронный ресурс]. – Режим доступа: <https://arduinoplus.ru/plata-arduino-uno/>
- 6 Лидары [Электронный ресурс]. - Режим доступа: <https://learnc.info/blog/lidars.html>
- 7 LED - электро [Электронный ресурс]. – Режим доступа: <http://led-elektro.ru/upravlenie-svetom/datchik-dvizheniya--tdm-mikrovolnovyyddm-01-58ggc1200vt10-720s1-8m>
- 8 Использование методов теории автоматического управления при разработке мехатронных систем [Электронный ресурс]. – Режим доступа: <http://window.edu.ru/resource/151/77151/files/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D1%8B%20%D0%A2%D0%90%D0%A3%20%D0%B2%20%D0%BC%D0%B5%D1%85%D0%B0%D1%82%D1%80%D0%BE%D0%BD%D0%B8%D0%BA%D0%B5.pdf#1>
- 9 Бесекерский, В.А. Теория систем автоматического регулирования / В.А. Бесекерский, Е.П. Попов – Москва: Профессия, 2004. – 747 с.

- 10 Теория автоматического управления: учебник для вузов: в 2 ч. / под ред. А.А. Воронова. - Москва: Высшая школа, 1986. – 504 с.
- 11 Лукас, В.А. Теория автоматического управления: учебник для вузов / В.А. Лукас. - Москва: Недра, 1990. – 416 с.
- 12 Теория автоматического управления: учебник / В.Н. Брюханов [и др.] ; под ред. Ю.М. Соломенцева. - Москва: Высшая школа, 2000. – 268 с.
- 13 Федотов, А.В. Анализ и синтез систем автоматического регулирования при проектировании средств автоматизации: учебное пособие / А.В. Федотов. - Омск: Издательство ОмГТУ, 1995. – 48 с.
- 14 Куропаткин, П.В. Теория автоматического управления / П.В. Куропаткин. – Москва: Высшая школа, 1973. – 528 с.
- 15 Егоров, К.В. Основы теории автоматического регулирования: учебное пособие для вузов / К.В. Егоров. - Москва: Энергия, 1967. – 648 с.
- 16 Справочник по теории автоматического управления / под ред. А.А. Красовского. – Москва: Наука, 1987. – 712 с.
- 17 Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/C%2B%2B>
- 18 Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Java>
- 19 Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Python>
- 20 Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Arduino>

ПРИЛОЖЕНИЕ А

Программный код для программирования системы

```
#define PIN_TRIG 12
#define PIN_ECHO 11

long duration, cm;

void setup() {
  // Инициализируем взаимодействие по последовательному порту
  Serial.begin (9600);
  //Определяем входы и выходы
  pinMode(PIN_TRIG, OUTPUT);
  pinMode(PIN_ECHO, INPUT);
}

void loop() {
  // Сначала генерируем короткий импульс длительностью 2-5
  микросекунд.
  digitalWrite(PIN_TRIG, LOW);
  delayMicroseconds(5);
  digitalWrite(PIN_TRIG, HIGH);
  // Выставив высокий уровень сигнала, ждем около 10 микросекунд. В
  этот момент датчик будет посылать сигналы с частотой 40 КГц.
  delayMicroseconds(10);
  digitalWrite(PIN_TRIG, LOW);
  // Время задержки акустического сигнала на эхолокаторе.
  duration = pulseIn(PIN_ECHO, HIGH);
  // Теперь осталось преобразовать время в расстояние
  cm = (duration / 2) / 29.1;

  Serial.print("Расстояние до объекта: ");
```



```

Serial.print(cm);
Serial.println(" см.");
// Задержка между измерениями для корректной работы скетча
delay(250);
}
#include <NewPing.h>

#define PIN_PING 12 // Пин с Arduino соединен с пинами trigger и echo
на датчике расстояния
#define MAX_DISTANCE 200 // Максимальное расстояние, которое мы
способны контролировать (400-500см).

NewPing sonar(PIN_PING, PIN_PING, MAX_DISTANCE); //
Регулировка пинов и максимального расстояния
void setup() {
Serial.begin(9600); // Открывается протокол с данными и частотой
передачи 115200 бит/сек.
}
void loop() {
delay(50); // Задержка в 50 мс между генерируемыми волнами. 29 мс –
минимально допустимое значение
unsigned int distanceSm = sonar.ping(); // Создание сигнала, получение
параметра его продолжительности в мкс (uS).
Serial.print("Ping: ");
Serial.print(distanceSm / US_ROUNDTRIP_CM); // Пересчет параметра
времени в величину расстояния и вывод результата (0 соответствует
выходу за допустимый предел)
Serial.println("cm");
}
#include <NewPing.h>

```

```

#define PIN_PING 12 // Пин с Arduino соединен с пинами trigger и echo
на датчике расстояния
#define MAX_DISTANCE 200 // Максимальное расстояние, которое мы
способны контролировать (400-500см).

NewPing sonar(PIN_PING, PIN_PING, MAX_DISTANCE); //
Регулировка пинов и максимального расстояния

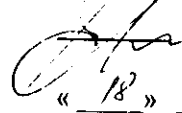
void setup() {
Serial.begin(9600); // Открывается протокол с данными и частотой
передачи 115200 бит/сек.
}
void loop() {
delay(50); // Задержка в 50 мс между генерируемыми волнами. 29 мс –
минимально допустимое значение
unsigned int distanceSm = sonar.ping(); // Создание сигнала, получение
параметра его продолжительности в мкс (uS).
Serial.print("Ping: ");
Serial.print(distanceSm / US_ROUNDTRIP_CM); // Пересчет параметра
времени в величину расстояния и вывод результата (0 соответствует
выходу за допустимый предел)
Serial.println("cm");
}

```

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра «Системы автоматики, автоматизированное управление и
проектирование»

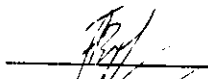
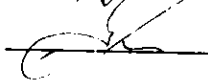
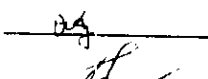
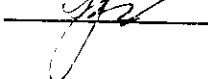
УТВЕРЖДАЮ
Заведующий кафедрой
С.В. Ченцов


« 18 » 06 2021 г.

БАКАЛАВРСКАЯ РАБОТА

27.03.04 «Управление в технических системах»

**СИСТЕМА ПРЕДОТВРАЩЕНИЯ СТОЛКНОВЕНИЙ
ЭЛЕКТРОСАМОКАТА**

Руководитель		18.06.2021 г.	ст. преподаватель П. В. Авласко
Консультант		18.06.2021 г.	С. В. Ченцов
Выпускник		18.06.2021 г.	А. А. Огородников
Нормоконтролер		18.06.2021 г.	Т. А. Грудинова

Красноярск 2021