

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

Космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ  
Заведующий кафедрой

О.В. Непомнящий

подпись                      инициалы, фамилия

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.01 Информатика и вычислительная техника

код и наименование направления

Моделирование и прототипирование микроконтроллерной системы круиз-  
контроля автомобиля

тема

Руководитель

\_\_\_\_\_

подпись, дата

доцент, канд. техн. наук

должность, ученая степень

В.Г. Середкин

инициалы, фамилия

Выпускник

\_\_\_\_\_

подпись, дата

Н.О. Красильников

инициалы, фамилия

Нормоконтролер

\_\_\_\_\_

подпись, дата

В.И. Иванов

инициалы, фамилия

Красноярск 2018

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

Космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ  
Заведующий кафедрой

О.В. Непомнящий

подпись                      инициалы, фамилия

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

**ЗАДАНИЕ  
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ  
в форме бакалаврской работы**

Студенту \_\_\_\_\_ Красильникову Никите Олеговичу  
фамилия, имя, отчество

Группа КИ14-08Б \_\_\_\_\_ Направление (специальность) 09.03.01  
номер код

«Информатика и вычислительная техника»

наименование

Тема выпускной квалификационной работы Моделирование и прототипирование микроконтроллерной системы круиз-контроля автомобиля

Утверждена приказом по университету № \_\_\_\_\_ от \_\_\_\_\_

Руководитель ВКР В.Г. Середкин, канд.техн.наук, доцент кафедры ВТ  
инициалы, фамилия, должность, учебное звание и место работы

Исходные данные для ВКР: разработать прототип системы круиз-контроля автомобиля на базе микроконтроллера с характеристиками приемлемыми для прототипирования системы, обеспечивающий реализацию функций:  
1 Поддержания заданной скорости; 2 Контроля критического расстояния до впереди идущего автомобиля; 3 Оповещения водителя при возникновении аварийно-опасной ситуации на пути следования автомобиля и необходимости экстренного торможения

Перечень разделов для ВКР: 1 Анализ задания на ВКР; 2 Разработка структурной схемы прототипа системы; 3 Разработка функциональной схемы прототипа системы; 4 Разработка электрической принципиальной схемы прототипа системы; 5 Разработка программного обеспечения для прототипа системы (схема алгоритма и базовые программные модули); 6 Тестирование аппаратно-программной части прототипа системы в выбранной среде моделирования

Перечень графического материала: презентация доклада выступления

Руководитель ВКР \_\_\_\_\_

подпись

В.Г.Середкин  
инициалы, фамилия

Задание принял к исполнению \_\_\_\_\_

подпись

Н.О.Красильников  
инициалы, фамилия

«\_\_» \_\_\_\_\_ 2018 г.

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Моделирование и прототипирование микроконтроллерной системы круиз-контроля автомобиля» состоит из: 67 страниц текста, 24 иллюстраций, 7 таблиц и 25 использованных источников.

**МИКРОКОНТРОЛЛЕРНАЯ СИСТЕМА УПРАВЛЕНИЯ КРУИЗ-КОНТРОЛЯ АВТОМОБИЛЯ, КОНТРОЛЬ КРИТИЧЕСКОГО РАССТОЯНИЯ, ОПОВЕЩЕНИЕ ОБ ЭКСТРЕННОМ ТОРМОЖЕНИИ.**

Объект разработки – Прототип микроконтроллерной системы круиз-контроля автомобиля с поддержкой функций контроля критического расстояния и оповещения об экстренном торможении.

Задачи:

- 1 Выполнение анализа существующих систем круиз-контроля автомобиля согласно задания на ВКР;
- 2 Разработка структурной схемы прототипа системы круиз-контроля автомобиля;
- 3 Разработка функциональной схемы прототипа системы круиз-контроля автомобиля;
- 4 Выбор компонентов прототипа системы круиз-контроля автомобиля;
- 5 Разработка схемы электрической принципиальной прототипа системы круиз-контроля автомобиля;
- 6 Разработка программного обеспечения для прототипа системы круиз-контроля автомобиля;
- 7 Тестирование аппаратно-программной части прототипа системы в выбранной среде моделирования;

В результате выполнения работы, разработана схема электрическая принципиальная прототипа системы круиз-контроля автомобиля и проведено тестирование аппаратно-программной составляющей прототипа в выбранной среде моделирования.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Анализ задания на ВКР.....	6
1.1 Цели и задачи ВКР.....	7
1.2 Обзор существующих аналогов.....	9
1.2.1 Active Cruise Control.....	10
1.2.2 Adaptive Cruise Control.....	15
Выводы к разделу 1.....	17
2 Разработка структурной схемы прототипа системы.....	19
Выводы к разделу 2.....	22
3 Разработка функциональной схемы прототипа системы.....	23
3.1 Выбор компонентов системы.....	23
3.1.1 Выбор микроконтроллера.....	24
3.1.2 Выбор датчика расстояния.....	25
3.1.3 Выбор датчика скорости.....	28
3.1.4 Выбор индикатора аварийной ситуации.....	29
3.1.5 Выбор функциональных клавиш.....	30
3.1.6 Выбор исполнительного устройства.....	31
3.1.7 Выбор источника питания.....	34
3.2 Разработка функциональной схемы прототипа.....	35
Выводы к разделу 3.....	36
4 Разработка принципиальной схемы прототипа системы.....	37
Выводы к разделу 4.....	39
5 Разработка программного обеспечения прототипа системы.....	40
5.1 Разработка граф-схемы алгоритма функционирования прототипа системы.....	40
5.2 Обзор средств разработки и тестирования ПО.....	45
5.2.1 AVR Studio 4.....	45
5.2.2 Code Vision AVR.....	46

Выводы к разделу 5.....	46
6 Тестирование аппаратно-программной части прототипа системы в выбранной среде моделирования.....	47
6.1 Обзор средств моделирования.....	47
6.1.1 Proteus.....	47
6.1.2 NI Multisim.....	48
6.2 Тестирование аппаратно-программного решения.....	49
Выводы к разделу 6.....	51
ЗАКЛЮЧЕНИЕ .....	52
СПИСОК СОКРАЩЕНИЙ.....	53
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	54
ПРИЛОЖЕНИЕ А.....	57

## ВВЕДЕНИЕ

Разработка микроконтроллерной системы круиз-контроля автомобиля, имеющей дополнительный функционал определения критического расстояния до препятствия и заблаговременного торможения перед ним, является одной из актуальных инженерных задач, поскольку ежедневно ведется разработка более качественных и устойчивых к воздействию факторов окружающей среды – компонентов элементной базы, имеющих потенциальную возможность реализации в разрабатываемых системах круиз-контроля нового поколения, превосходящих предшествующие аналоги.

Основными отличительными характеристиками разрабатываемых систем круиз-контроля нового поколения являются:

- 1 Высокий уровень безопасности, достижимый за счет применения соответствующих вспомогательных систем и проработанных программных алгоритмов;
- 2 Повышенный рабочий температурный диапазон компонентов системы;
- 3 Эргономика и информативность интерфейса пользователя.

В настоящее время система круиз контроля, это вспомогательная система автомобиля, основной задачей которой является поддержание заданной скорости движения, вне зависимости от условий движения, будь то: движение с отрицательным (с горы) или положительным (в гору) углом по отношению к дорожному покрытию, движение по автодорогам магистрального или городского типа.

Возможность задания и поддержания скорости движения автомобиля полезна в случае долгих поездок, когда в течение длительного времени человеку приходится совершать статичное усилие для поддержания оптимальной скорости, путем нажатия на педаль акселератора. Помимо этого, за счет интуитивного использования ресурсов автомобиля, система



обеспечивает топливную экономию и повышенный срок службы рабочих узлов автомобиля.

Интерфейс управления системой круиз-контроля может быть реализован в разных интерпретациях: в виде многофункционального рычага, дополнительного блока клавиш управления на руле или на приборной панели автомобиля.

Системы круиз-контроля подразделяют на четыре типа: ограничители скорости, пассивный круиз контроль, адаптивный круиз контроль, полуавтономный круиз-контроль [1].

- Ограничитель скорости;

Система фиксирует значение максимальной скорости и не дает водителю превысить заданный лимит.

- Стандартный круиз-контроль;

Система, имеет функцию установки постоянной скорости движения транспортного средства и реагирует на изменение разницы между действующей скоростью и заданной.

- Адаптивный круиз контроль;

Является более передовым решением поскольку оснащено дополнительным функционалом, таким как: регулирование оптимальной скорости в динамике, автоматическое экстренное торможение, контроль критического расстояния и т.д.

- Полуавтономный круиз контроль.

В сравнении с другими аналогами данный тип систем круиз-контроля включает в себя помимо всех перечисленных функций адаптивного круиз контроля, более сложные подсистемы, которые самостоятельно могут вмешиваться в рулевое управление автомобиля.

## 1 Анализ задания на ВКР

Определим этапы анализа задания на ВКР для обеспечения планомерного и детального обзора поставленных задач.

В первую очередь, необходимо выяснить какая цель поставлена перед исполнителем в задании на ВКР, чтобы обеспечить неукоснительное выполнение всех заданных требований.

Поскольку в рамках задания на ВКР требуется произвести разработку прототипа заданной системы, необходимо выполнить сравнительный обзор аналогов системы, для дальнейшего выявления отличительных особенностей и недостатков. Результат анализа систем-аналогов учесть при разработке прототипа.

При дальнейшем анализе задания на ВКР необходимо провести обзор:

- ❖ Аппаратной части, для разработки функциональной схемы прототипа;
- ❖ Элементов электрической цепи, для разработки электрической принципиальной схемы прототипа;
- ❖ Среды разработки ПО, для разработки и тестирования ПО прототипа;
- ❖ Среды моделирования, для совместного моделирования аппаратно-программной части прототипа и выявления работоспособности разработанного решения.

Следует заметить, что для обеспечения планомерного анализа задач на ВКР, обзор каждого конкретного элемента задачи, включенного в вышеуказанное перечисление, требуется непосредственно при разработке соответствующего раздела ВКР.

Следовательно, анализ аппаратной части будет проведен непосредственно в разделе – «Разработка функциональной схемы прототипа системы», анализ элементов электрической цепи будет проведен в разделе «Разработка электрической принципиальной схемы системы» и т.д.

## 1.1 Цели и задачи ВКР

Целью выпускной квалификационной работы является, разработка прототипа МК-системы круиз-контроля с функциями: поддержания заданной скорости, контроля критического расстояния до препятствия, оповещении водителя в случае аварийно-опасной ситуации на пути следования и необходимости экстренного торможения.

Обозначим и охарактеризуем основные функции требующие реализации в разрабатываемом прототипе микроконтроллерной системы круиз-контроля:

- Фиксация скорости движения;

Предполагает возможность считывания с акселерометра действующей скорости автомобиля и задания её при помощи нажатия на клавишу «SET» на приборной панели системы круиз контроля, при включенном состоянии системы.

- Поддержание заданной скорости;

Производится путем одновременного снятия показаний с датчика положения дроссельной заслонки (ДПДЗ) и датчика скорости автомобиля (ДС), и соизмерении микроконтроллером угла открытия дроссельной заслонки (ДЗ) с разницей действующей скорости автомобиля (ДСА) и эталонной скорости автомобиля (ЭСА) [2].

Регулировка положения дроссельной заслонки производится при помощи двух исполнительных устройств: исполнительное устройство дроссельной заслонки (ИУДЗ) и исполнительного устройства тормозной системы (ИУТС). ИУДЗ работает в случае набора скорости автомобиля до ЭСА, ИУТС работает в случае сброса скорости до ЭСА [2].

- Контроль критического расстояния;

Контроль критического расстояния является комплексной функцией системы круиз-контроля, состоящей из нескольких подфункций, таких как:

- 1 Автоматическое торможение перед неподвижным препятствием;

Согласно ЭСА и расстоянию до неподвижного препятствия измеряемое при помощи датчика расстояния (ДР), система высчитывает величину остановочного пути (ОП), необходимую для расчета усилия на педаль тормоза при помощи ИУТС.

ИУТС соизмеримо действует на педаль тормоза и приводит ДЗ в положение открытия при холостом ходе (ХХ), соответственно топливно-воздушная смесь (ТВС) поступает в объеме необходимом для поддержания двигателя в рабочем состоянии, предотвращая тем самым дальнейший разгон автомобиля.

## 2 Автоматическое торможение в случае движущегося препятствия.

В данной ситуации, автомобиль двигаясь с ЭСА догоняет впередиидущий автомобиль. Для обеспечения безопасного преследования автомобиля, система должна скорректировать ЭСА согласно скорости впередиидущего автомобиля. Расчет новой ЭСА производится с помощью ДР, поскольку он фиксирует автомобиль на расстоянии меньшем чем ОП, система управления снижает или увеличивает значение заданной водителем скорости согласно условию, обозначенному в программном коде системы.

При резком снижении скорости впередиидущего автомобиля, система оповещает водителя о возможном возникновении аварийной ситуации при дальнейшем преследовании и подает соответствующий звуковой сигнал при помощи звукового индикатора, для своевременного вмешательства водителя в управление автомобилем.

- Оповещение об экстренном торможении.

При возникновении внезапной преграды на пути следования автомобиля система подает соответствующий сигнал водителю с помощью звукового индикатора. Подача сигнала происходит в том случае если внезапная преграда возникла на расстоянии меньшем чем ОП [2].

## 1.2 Обзор существующих аналогов

Для дальнейшего анализа принципа работы рассматриваемой системы, проведем обзор аналогов максимально схожих по функционалу с разрабатываемым по заданию ВКР – прототипом.

Требуется, чтобы системы-аналоги поддерживали такие функции как:

- Движение с заданной скоростью;
- Контроль критического расстояния;
- Оповещение об аварийно-опасной ситуации.

Так же известно, что совокупность приведенного функционала, удовлетворяющего требованиям поиска, относится к типу активного круиз-контроля, соответственно ведем поиск аналогов среди данного типа систем.

После некоторого времени поиска, определены системы-аналоги, которые схожи по характеру выполняемых функций с разрабатываемым прототипом системы круиз-контроля.

Рассмотрим, в качестве аналогов разрабатываемого прототипа, системы активного круиз-контроля такие как – Active Cruise Control и Adaptive Cruise Control [3].

Проведем подробный обзор аналогов и выясним: основные рабочие узлы систем, характеристики аппаратных компонентов, принцип и условия работы систем. В конце главы проведем сравнительный анализ систем аналогов и выделим полезные для разработки прототипа моменты.

## 1.2.1 Active Cruise Control

Рассмотрим один из аналогов систем адаптивного являющегося разработкой компании и названием системы – (SA541) Active Cruise Control ver.2. (ACC2). Основная графическая схема функционирования системы представлена на рисунке 1.

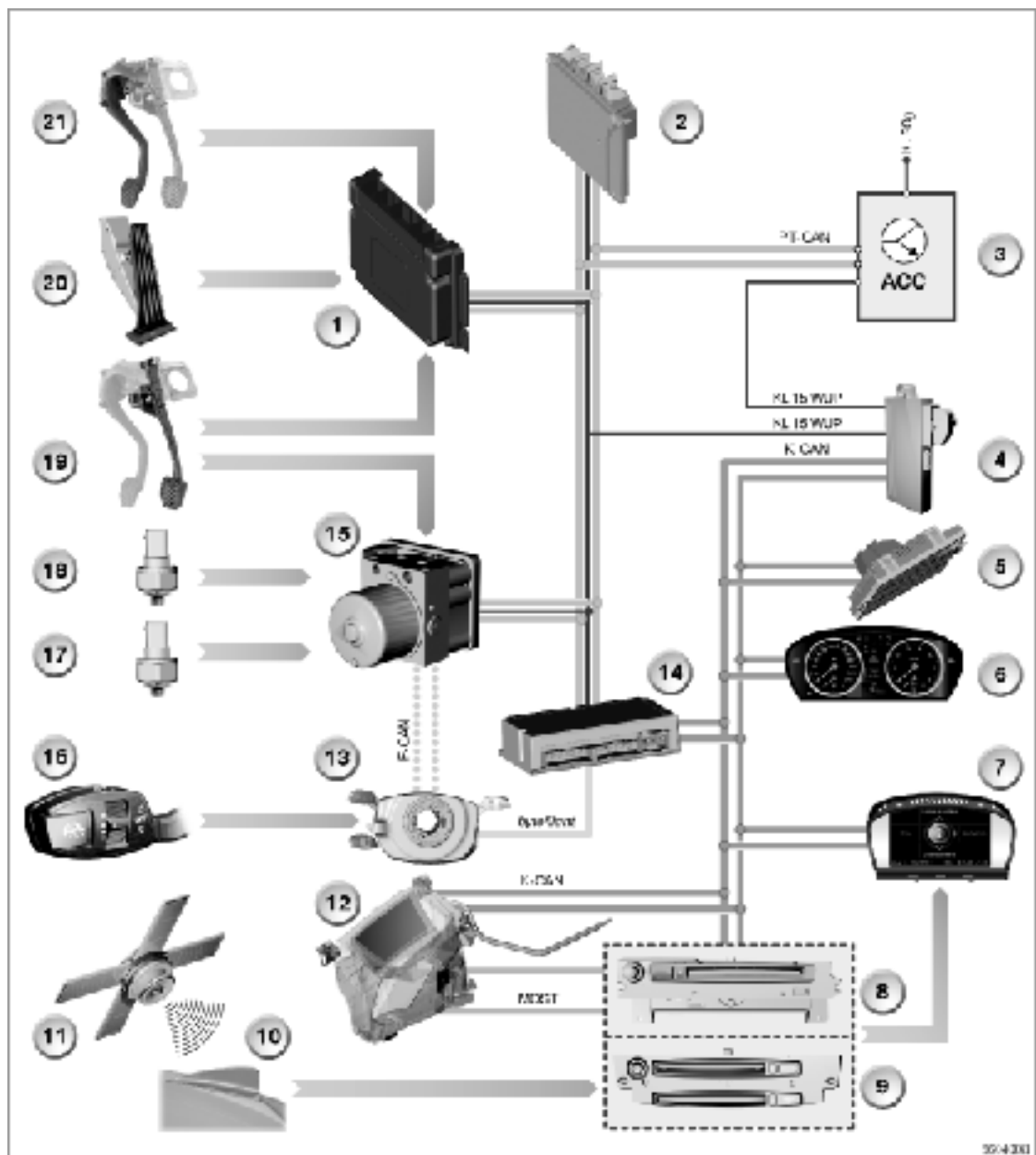


Рисунок 1 – Active Cruise Control (ACC2)

Определим основные характеристики компонентов аппаратной части аналога, его функциональные подсистемы и принципы работы системы:

1 Компоненты аппаратной части;

- Датчик АСС с блоком управления;

Датчик АСС и блок управления представляют собой единый узел. Датчик АСС определяет расстояние, угол и относительную скорость впереди идущих автомобилей (Относительная скорость: относительная скорость - это разность скоростей двух движущихся объектов.).

- Переключатель круиз-контроля;

Управление системой активного круиз-контроля осуществляется с помощью выключателя круиз-контроля на рулевой колонке.

- Датчики тормозного давления.

По одному датчику тормозного давления встроено в контуры тормозного привода передних и задних колес. Сигналы датчиков тормозного давления анализируются системой динамического контроля стабильности.

При активированном круиз-контроле автомобиль при необходимости подтормаживается системой курсовой устойчивости (Dynamic Stability Control (DSC)). С помощью датчиков тормозного давления осуществляется регулировка зачастую лишь незначительного тормозного давления передних и задних колес [4].

2 Функциональные подсистемы;

- Система динамического контроля стабильности (DSC – Dynamic Stability Control);

Режим движения автомобиля распознается системой DSC на основании анализа следующих сигналов датчиков:

- скорость вращения автомобиля вокруг вертикальной оси;
- угол поворота рулевого колеса
- окружная скорость колес

От DSC данные о текущем режиме движения передаются на датчик АСС.

От датчика ACC на DSC передается команда на снижение скорости автомобиля: ACC передает на DSC команду снижения скорости автомобиля по шине PT-CAN. С помощью датчиков тормозного давления, включенных в состав системы DSC, осуществляется незначительное тормозное давление на передние и задние колеса.

- Цифровая электронная система управления двигателем (DME);

DME принимает запрос от блока управления ACC на измерение крутящего момента, в ответ на запрос DME передает сигнал положения модуля педали акселератора.

- Модуль безопасности и межсетевое преобразование (SGM);

По своей сути является интерфейсом передачи данных между шинами PT-CAN, K-CAN, byteflight.

- Модуль педали акселератора (FPM);

Модуль педали акселератора передает сигнал с задаваемыми водителем значениями. Этот сигнал необходим для проверки соответствующего режима движения. В результате нажатия на педаль акселератора в любой момент можно увеличить значение скорости, заданное для активного круиз-контроля.

- Выключатель стоп-сигналов;

Выключатели стоп-сигналов передают в систему АКК сигнал о нажатии педали тормоза. При торможении АКК выключается.

- Электронная система управления коробкой передач (EGS);

Система EGS анализирует данные от DME/DDE и датчика ACC.

EGS система является составной частью ЭБУ коробки передач и согласует характер переключения программ движения с задаваемыми водителем значениями и дорожной обстановкой, путем обмена информацией о включенной передаче и о ранних переключениях передач, с датчиком ACC.

- Световой модуль (LM);

Для оповещения об автоматическом торможении системой круиз-контроля необходима передача сигнала на световой модуль по шине K-CAN.



В ситуации смены полосы движения, активный круиз-контроль использует сигнал указателя поворота от светового модуля, для того чтобы перед обгоном, после нажатия левого указателя поворота, система как можно быстрее перестала фиксировать обгоняемый автомобиль и наоборот как можно быстрее распознавала автомобиль впередиидущий после перестроения в правый ряд.

- Коммутационный центр в рулевой колонке (SZL);

Обрабатывает все данные от переключателя круиз-контроля на рулевой колонке. Через шину «byteflight» центр соединен с модулем SGM, который осуществляет соединение с остальными шинами системы через PT-CAN шину.

- Многофункциональный контроллер аудиосистемы (M-ASK/CCC).

Служит в качестве интерфейса оповещения. Подает через динамик звуковой предупреждающий сигнал в аварийно-опасной ситуации для оповещения водителя. Управление системой производится через шину K-CAN

Датчик АСС с блоком управления установлен в передней части автомобиля под передним бампером. Более детально изучить конструкцию датчика АСС можно на рисунке 2.

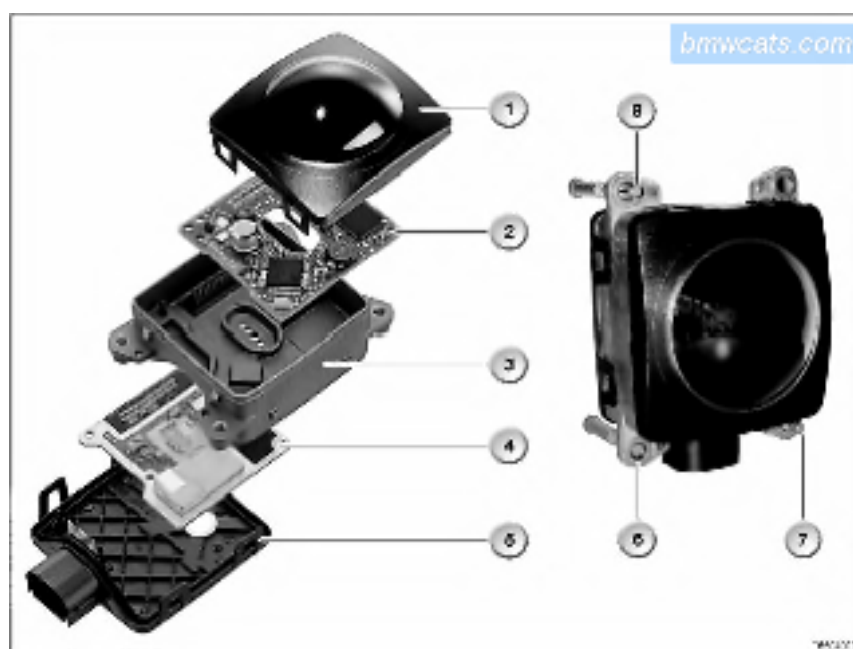


Рисунок 2 – Датчик АСС с блоком управления (БУ)

Проанализировав конструкцию выделим что датчик АСС и блок управления представляют собой единый узел и состоит из 8 элементов. Из них нас интересуют элементы – (2) Плата, (4) Плата с электронными элементами высокой частоты и (5) Задняя часть корпуса с разъемом и элементом для выравнивания давления.

Рассматривая принцип действия АСС заметим, что активизировать систему можно только при скорости не менее 30 км/ч. Датчик АСС всегда включается при скорости около 15 км/ч (готовность к активизации). Благодаря этому сразу после включения круиз-контроля возможна регулировка.

Датчик АСС обследует зону действия перед автомобилем с помощью 4 конусов радиолокационного излучения. Принцип действия радиолокационного датчика можем увидеть на рисунке 3.

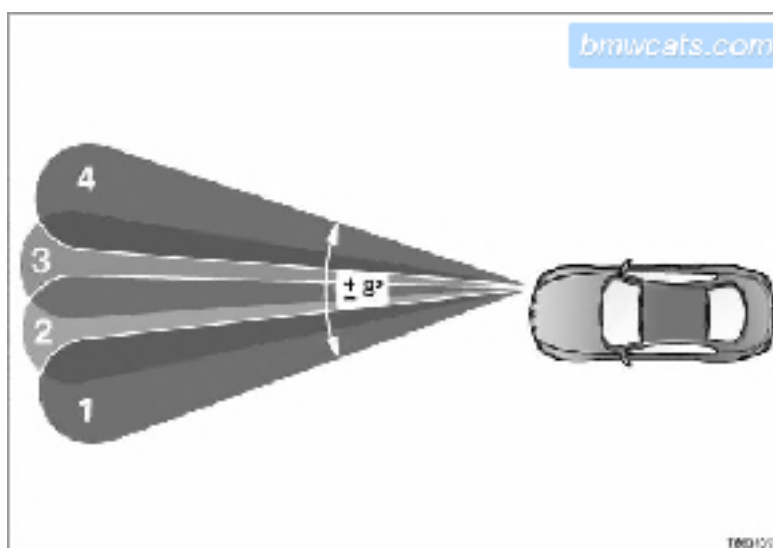


Рисунок 3 – Принцип действия радиолокационного датчика

Каждый конус радиолокационного излучения генерируется одной антенной, которая и принимает отраженный радиолокационный сигнал.

Диапазон измерения от 2 до 120 метров. Частота излучения датчика АСС около 76 ГГц. Благодаря этому осуществляется постоянный контроль зоны

действия. Зона действия ограничена по сторонам и по высоте углом раствора датчика АСС.

Измерение дистанции и относительной скорости Блок управления АСС производит следующим образом:

- Дистанция рассчитывается по времени распространения частотных спектров;

- Относительная скорость рассчитывается на основе эффекта Доплера. Эффект Доплера является законом физики: Если объект приближается к передатчику, частота отраженного радиолокационного сигнала увеличивается. Если объект удаляется от передатчика, частота отраженного радиолокационного сигнала уменьшается [3].

### 1.2.2 Adaptive Cruise Control

Для дальнейшего анализа рассмотрим систему АСС с функцией stop & go («старт-стоп»), разработанную компанией Audi. Принцип функционирования системы можно увидеть на рисунке 4.

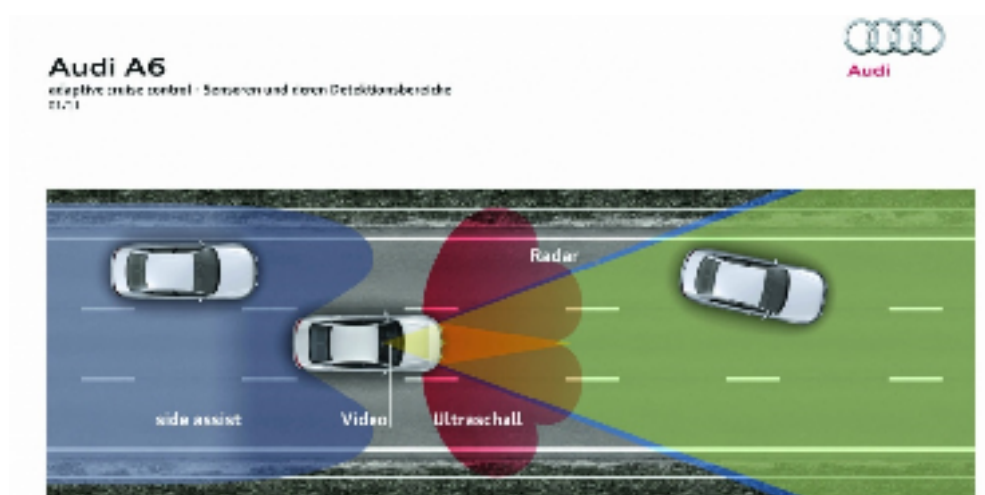


Рисунок 4 – Принцип функционирования системы АСС stop & go

Система ACC stop & go использует два радиолокационных датчика представленные на рисунке 5. Датчики установлены в передней части автомобиля и имеют систему обогрева для предотвращения запотевания линзы датчика. Радиолокационные датчики работают с частотой 76,5 ГГц, измерения датчиков ограничены диапазоном угла в 40 градусов и максимальным расстоянием фиксации до 250 м. Компьютер обрабатывает сигналы с датчиков таким образом обнаруживая впереди идущие транспортные средства [5].



Рисунок 5 – Радиолокационный датчик системы ACC stop & go

Водитель имеет возможность управлять заданным интервалом до впереди идущего транспортного средства и динамикой разгона автомобиля.

Функционал системы ACC stop & go построен на взаимодействии с данными других информационных систем, имеющих около 30 БУ для более точного анализа пространства вокруг автомобиля, полученная информация позволяет системе распознавать сложные сценарии и прогнозировать поддержку водителя. Поскольку система так же синхронизирует данные с навигационными картами, расчет курса заранее выбранного маршрута помогает определить движение по дугообразной траектории.

## Выводы к разделу 1

По итогу раздела «Анализ задания на ВКР», в котором произведена постановка общей цели и задач, составлен поэтапный план для достижения поставленной задачи по разработке прототипа системы круиз-контроля автомобиля.

Так же, произведен обзор аналогов схожих по функционалу с разрабатываемым прототипом, соответственно появилась возможность провести сравнение характеристик и отметить ключевые моменты перед дальнейшей разработкой прототипа. Отличительные особенности систем представлены в виде таблицы 1 [4, 5].

Таблица 1 – Сравнительные характеристики систем-аналогов круиз-контроля

Название системы	Active Cruise Control (BMW)	Adaptive Cruise Control (AUDI)
Диапазон задания скорости, км/ч	От 30 до 180	От 40 до 220
Тип датчика расстояния	Радиолокационный	Радиолокационный
Частота измерения датчика скорости, ГГц	76	76,5
Дальность измерения датчика расстояния, м	До 120	До 250
Угол раствора радиолокационного излучения, °	До 32	До 40
Принцип задействования торможения до впередиидущего автомобиля	Водитель задает временной период, через который производится фиксация расстояния до автомобиля. Система производит торможение при помощи сигнала с ЭБУ на тормозную систему автомобиля	В системе фиксированный таймаут считывания информации датчиком, его значение зависит от отношения действующей скорости и расстояния до впередиидущего автомобиля. Система производит торможение при помощи сигнала на ответственную систему, через ЭБУ
Система оповещения при аварийно-опасных ситуациях	Воспроизводится звуковой сигнал при помощи контроллера аудиосистемы и выводится оповещение на дисплей.	Воспроизводится звуковой сигнал при помощи контроллера аудиосистемы и выводится оповещение на дисплей.

Из полученных данных следует, что для реализации задач, требуемых к разрабатываемому прототипу, необходимо обширное количество подсистем, регулирующих тонкие аспекты во время движения автомобиля. Это усложняет процесс разработки и соответственно повышает общую стоимость подобных систем. Соответственно, при реализации прототипа необходимо руководствоваться выполнением поставленных задач, упрощением компонентной базы системы и рассмотрении прототипа в качестве системы реализуемой только внутри разрабатываемой модели.

## 2 Разработка структурной схемы прототипа системы

Согласно разрабатываемой системы круиз-контроля автомобиля, для более детального обзора, составим структурную схему прототипа, включающую в себя: пояснение реализованных функций и принципы взаимодействия аппаратных составляющих системы.

Включение системы и установка скоростного режима производится соответственно при помощи функциональных клавиш – «ON/OFF» и «SET». При нажатии на клавишу «ON/OFF» система переходит в активное состояние или отключается, производя сброс системы. При включении подается сигнал на LED – элемент, который оповещает водителя о состоянии включения системы. При нажатии функциональной клавиши «SET», в блок обработки подается сигнал на считывание скорости с ДС, для установки её значения как эталонного.

Во время движения автомобиля с заданной скоростью, система фиксирует показания с ДС и сравнивает показатели с заданными. В случае если скорость автомобиля меньше заданной, блок вывода подает сигнал на ИУ1 который постепенно открывает дроссельную заслонку, когда заданная скорость достигнута, положение дроссельной заслонки, при достижении требуемой скорости остается неизменным.

При спуске с горы автомобиль набирает скорость большую заданной, соответственно нужно уменьшить степень угла открытия дроссельной заслонки и произвести торможение. С этими функциями справляется ИУ2. ИУ2 воздействует на педаль тормоза соизмеримо величине полученного сигнала с блока ввода/вывода и обработки. От степени нажатия на педаль регулируется давление ТЖ внутри ТСА, так же стоит учесть, что большинство автомобилей нижнего ценового сегмента имеют аналоговую систему закрытия ДЗ реализованную в виде троса, напрямую соединяющего ось вращения ДЗ (в сторону закрытия) и педаль тормоза. Соответственно в разрабатываемой системе при использовании ИУ2, в ситуации «спуска с горы», регулирование

положения ДЗ производится автоматически аналоговым методом – после воздействия на педаль тормоза, соответственно система выполняет все мероприятия по понижению скорости до заданного значения.

Для поддержки функций фиксации препятствия и дальномера, в системе управления круиз-контролем использован датчик расстояния (ДР). Функция фиксации препятствий требуется для определения препятствия на пути следования автомобиля и возможности отследить его телеметрию. Функция дальномера важна для расчета степени усилия торможения при помощи ИУ2, поскольку в системе управления круиз-контролем величина ДСА известна, при помощи ДС, а расстояние до объекта при помощи ДР. Когда системе известно обе этих величины, возможен расчет величины ОП, зная которую система может рассчитать тормозное усилие для реализации автоматического плавного торможения [6].

Во время работы системы угол открытия ДЗ варьируется от уровня положения при ХХ до полностью открытой ДЗ (90°). Для того чтобы фиксировать положение в системе подключен ДПДЗ. Положение дроссельной заслонки в системе управления круиз-контролем автомобиля является изменяемой величиной на протяжении всего использования системы, соответственно система управления должна контролировать этот немаловажный показатель в реальном времени и при помощи отдельного датчика.

Для предотвращения аварийно-опасных ситуаций при внезапном появлении препятствия на пути следования автомобиля, на расстоянии меньшем чем ОП, система должна оповещать об этом водителя в экстренном порядке. Для предотвращения аварийных ситуаций, в рамках разрабатываемой системы используется звуковой индикатор аварийных ситуаций (ИАС), который своевременно подает информирующий звуковой сигнал для водителя.

В случае экстренной ситуации, когда требуется немедленно отключить систему управления и перейти к самостоятельному управлению автомобилем,



водитель в рамках разрабатываемой системы имеет возможность воспользоваться нажатием на педаль акселератора. В устройстве педали акселератора встроен датчик педали акселератора (ДПА) для экстренного отключения системы управления круиз-контролем по факту нажатия педали.

Для обеспечения питания системы управления круиз-контроля используется источник питания (ИП) напряжением в 12 В.

Снятие показаний, со всех примененных в функционале прототипа датчиков, кроме датчика расстояния, производится при помощи ЭБУ автомобиля с применением CAN интерфейса передачи данных.

Структурная схема прототипа системы круиз-контроля автомобиля представлена на рисунке 6.



Рисунок 6 – Структурная схема прототипа системы круиз-контроля автомобиля



## **Выводы к разделу 2**

В разделе 2 проведен более тщательный анализ технической составляющей автомобиля, выполнено словесное формулирование решений на основе анализа данных главы 1, в виде абстрактной модели, которая, в свою очередь, послужила основой для обозначения принципов взаимодействия аналоговых узлов автомобиля и цифровых узлов разрабатываемого прототипа. Разработана методология реализации прототипа системы и структурная схема прототипа.

### **3 Разработка функциональной схемы прототипа системы**

Во время разработки функциональной схемы прототипа системы управления круиз-контролем автомобиля, следует рассмотреть на предмет совместимости аналоги применяемых в дальнейшем систем компонентов, которые ранее обозначили при разработке структурной схемы в главе 2. Так же следует проанализировать их характеристики и построить на основе выбранных аппаратных решений – функциональную схему прототипа системы.

#### **3.1 Выбор компонентов системы**

Выбор компонентов для прототипа системы круиз-контроля автомобиля следует осуществить согласно составленной ранее структурной схемы. Необходимо учесть межкомпонентную совместимость, потребление и их применимость к разрабатываемому прототипу.

Определим список необходимых для выбора компонентов:

- 1 Микроконтроллер;
- 2 Датчик расстояния;
- 3 Датчик скорости;
- 4 Индикатор аварийной ситуации;
- 5 Функциональные клавиши;
- 6 Электродвигатели ИУ1 и ИУ2;
- 7 Источник питания системы.

### 3.1.1 Выбор микроконтроллера

По заданию необходимо выбрать микроконтроллер в соответствии с заданием на ВКР. Для обоснования выбора проведем сравнение микроконтроллеров по ряду параметров, и представим в таблице 2.

Таблица 2 – Сравнительная таблица микроконтроллеров фирмы Atmel

Микроконтроллер	AT90CAN32-Automotive	ATmega328	ATmega8535
FLASH, КБайт	32	32	8
EEPROM, байт	1024	1024	512
SRAM, байт	2048	2048	512
Кол-во портов I/O	53	23	32
Кол-во внешних источ. прерыв.	24	24	3
Таймеры	2 - 8разр., 1 - 16разр.	2 - 8разр., 1 - 16разр.	2 - 8разр., 1 - 16разр.
SPI	2	2	1
I2C	1	1	1
UART	1	1	1
АЦП (10 разр.)	6	8	8
Рабочая частота, МГц	0 ... 10 0 ... 20	0 ... 4 0 ... 10 0 ... 20	0 ... 16
Напряжение питания, В	2.7 ... 5.5	1.8 ... 5.5 2.7 ... 5.5 4.5 ... 5.5	4.5 ... 5.5
Кол-во DIP	28	28	40

Наиболее приемлемый для решаемой задачи является микроконтроллер AT90CAN32-Avtomotive, включающий автомобильный интерфейс CAN, но в силу его недоступности для лабораторных образцов используем в качестве элемента прототипирования микроконтроллер ATmega8535.

Из рассмотренных МК заметим, что для осуществления поставленной задачи МК ATmega 8535 является достаточным для разработки прототипа системы по ряду причин:

1 По сравнению с аналогами имеет необходимое кол-во портов ввода-вывода, это так же является плюсом при доработке системы;

2 Имеет достаточное, для реализации прототипа, кол-во портов с поддержкой АЦП, чем не могут;

3 Является более ресурсоемким и экономически выгодным для разрабатываемого прототипа системы.

На рисунке 7 изображен внешний вид микроконтроллера ATmega8535.

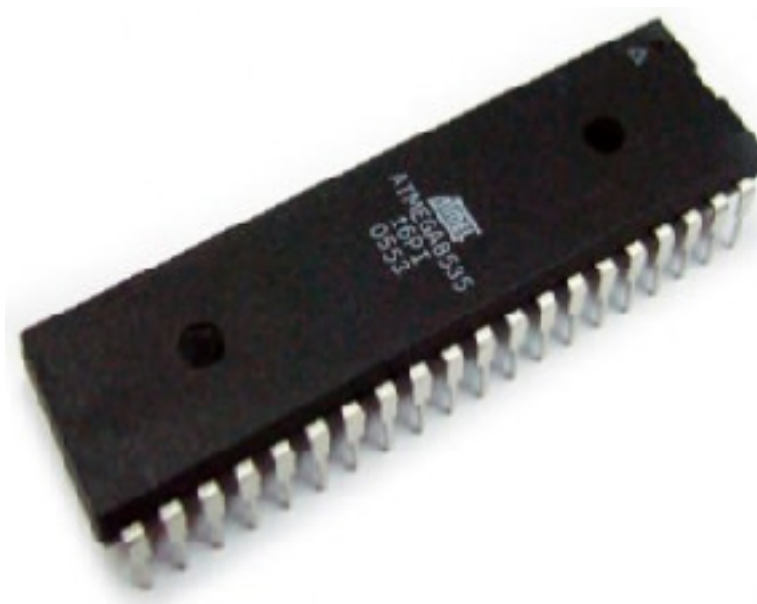


Рисунок 7 – Микроконтроллер ATmega8535

### 3.1.2 Выбор датчика расстояния

Главными критериями при выборе датчика расстояния согласно заданию ВКР являются: обширный диапазон расстояния, на котором производится фиксация препятствия на пути следования автомобиля, с максимальным значением, исчисляемым от 60 до 150+м.

При поиске датчика расстояния, удовлетворяющего параметрам дальности и точности определения препятствия, было найдено передовое решение датчика расстояния реализованного в виде микросхемы – IWR1443BOOST производства компании Texas Instruments. Внешний вид микросхемы представлен на рисунке 8.

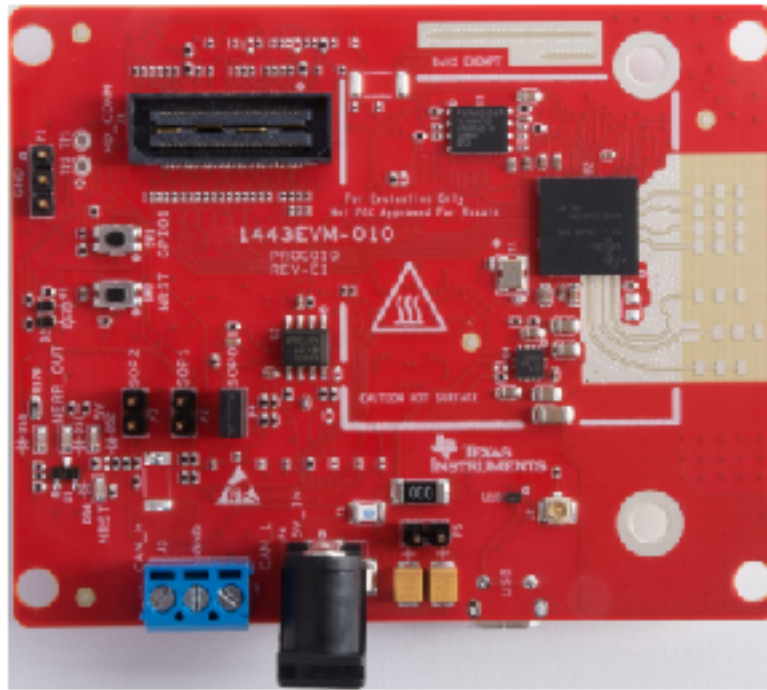


Рисунок 8 – Датчик расстояния IWR1443

Данный датчик является новинкой разработки среди существующих аналогов и как нельзя лучше подходит для применения при разработке систем круиз-контроля, поскольку демонстрирует непревзойденную точность, дальность измерения расстояния до 300 метров и высокую помехоустойчивость при неблагоприятных погодных условиях.

Так же выбор данной микросхемы в качестве датчика расстояния обусловлен тем, что микросхема является самостоятельной МК-системой которая может распознавать препятствие фиксировать расстояние до него и определять скорость впередиидущего объекта и объекта. на котором монтирован сам датчик. Это позволяет взамен датчика скорости получать информацию о скорости с данной микросхемы, это способствует оптимизации схемы прототипа и экономии при реализации прототипа.

Датчик является представителем микроволновых радаров с диапазоном от 76 до 81ГГц и напряжением питания 5В. Точность определения азимута датчика составляет единицы градусов, что достигается использованием

технологии ММО и фазированной антенной решетки на микрополосковых элементах печатной платы.

Главными критериями выбора именно этого датчика для реализации прототипа системы являются:

1 Наличие на микросхеме собственного микроконтроллера – ARM© R4F, позволяющего проводить все расчеты самостоятельно внутри схемы при этом не используя память микроконтроллера прототипа для обработки при подключении;

2 Микросхема помимо расчета расстояния так же измеряет действующую скорость автомобиля, это позволяет не производить выбор датчика скорости или подключение к датчику скорости автомобиля через ЭБУ.

3 Наличие UART интерфейса для подключения к ATmega8535 по которому производится передача информационного сигнала со скоростью до 100 кбит/с [8,9,10].

Несмотря на полную совместимость данного датчика с выбранным микроконтроллером, возможность приобретения данной микросхемы на момент прототипирования системы – недоступна. Так же не представляется возможным произвести тестирование данного датчика в средах моделирования поскольку его спецификация еще не занесена в библиотеки средств моделирования, поэтому необходимо использование более экономичного и совместимого с разрабатываемым прототипом – датчика, такого как ультразвуковой датчик расстояния – HC-SR04.

Параметров данного датчика хватает для проверки на работоспособность прототипа в средах моделирования. Внешний вид датчика представлен на рисунке 9.





Рисунок 9 – Ультразвуковой датчик расстояния HC-SR04

Данный датчик расстояния работает следующим образом: на выход TRIG подается сигнал высокого уровня, длительностью 10-15 микросекунд. После этого на вывод ECHO будет подан сигнал высокого уровня, длительностью пропорциональной измеренному расстоянию. То есть микроконтроллеру достаточно измерить длительность ECHO сигнала для определения расстояния до объекта. Для того чтобы перевести измеренное значение в сантиметры, нужно продолжительность импульса поделить на 58 [11].

### 3.1.3 Выбор датчика скорости

В функционале микросхемы IWR1443BOOST изначально числится возможность измерения действующей скорости автомобиля, но поскольку из-за невозможности тестирования компонента в средах моделирования, её использование исключено, соответственно вопрос о выборе датчика скорости остается открытым.

В конструкции автомобиля изначально используется датчик скорости, реализованный в виде датчика Холла, который подключен к ЭБУ по шине CAN, но поскольку при построении прототипа возможность приобретения МК с поддержкой CAN интерфейса недоступна, подключение к датчику скорости автомобиля так же не представляется возможным, соответственно необходимо

произвести выбор датчика, используемого отдельно от интерфейсов автомобиля.

Для данного прототипа системы как нельзя лучше подходит датчик SS459A, принцип его действия построен на элементе Холла и его достаточно для определения кол-ва оборотов вращения и соответственно для определения скорости. Напряжение питания датчика составляет 5В, диапазон рабочих температур составляет от  $-40^{\circ}$  до  $+150^{\circ}$ , это позволяет использовать датчик при большинстве погодных условий и для прямого подключения к выбранному микроконтроллеру ATmega8535 [12]. Внешний вид датчика скорости SS459A представлен на рисунке 10.

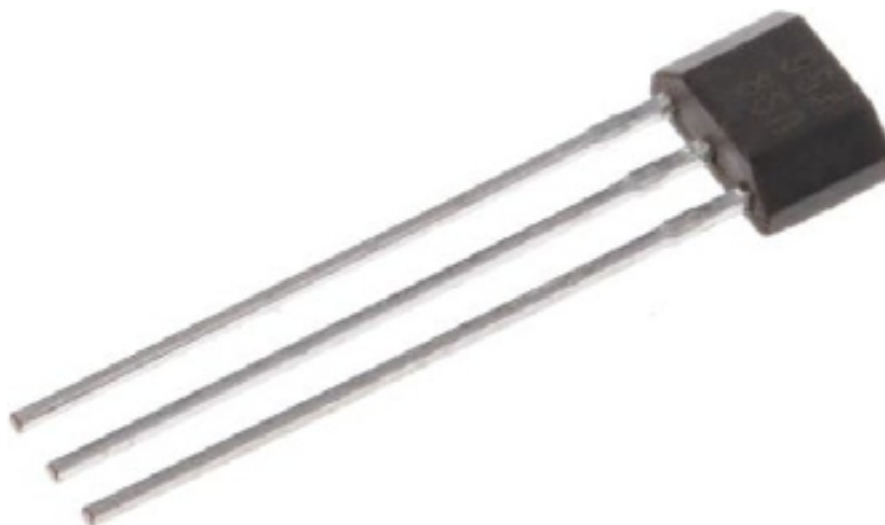


Рисунок 10 – Датчик скорости SS459A

#### **3.1.4 Выбор индикатора аварийной ситуации**

В качестве индикатора аварийной ситуации может служить как дисплей, так и звуковой сигнал. Дисплей может отображать какое-либо сообщение, но это плохо информирует о наличии аварийной ситуации. Звуковой сигнал хорошо подходит для информирования о наличии аварийной ситуации на

расстоянии. В таблице 2 представлены некоторые звуковые сигналы и их характеристики.

Таблица 2 – Характеристики звуковых сигналов

Название	Потребляемая мощность, Вт	Диапазон частот, Гц	Напряжение, В	Громкость, дБ
Звуковой сигнал M26	32	335	12	70
Звуковой сигнал DL50	32	310	12	75
Звуковой сигнал M100	16	350	12	73

Исходя из представленных звуковых сигналов в таблице 3 в качестве звукового индикатора аварийной ситуации подходит звуковой сигнал M100. Он обладает громкостью в 73 дБ и самой меньшей потребляемой мощностью в 16Вт. На рисунке 11 представлен внешний вид звукового сигнала M100 [13].



Рисунок 11 – Звуковой индикатор M100

### 3.1.5 Выбор функциональных клавиш

Функциональные кнопки необходимы для ввода номера стеллажа, отправки груза и сброса системы. Существует множество кнопок, и все они

похожи друг на друга. Отличаются они лишь формой и размером. В таблице 3 представлена сравнительная характеристика некоторых кнопок.

Таблица 3 – Характеристика функциональных кнопок

Название	Рабочее напряжение, В	Рабочий ток, А	Высота, мм	Типоразмер
KLS7-TS1204-7.3-180	12	0,05	7	12x12
KLS7-TS1202-4.3-180	12	0,05	4,3	12x12
KLS7-TS6601-4.3-180	12	0,05	4,3	6x6
KLS7-TS6601-11-180	12	0,05	11	6x6

Представленные кнопки в таблице 2 имеют одинаковое рабочее напряжение и рабочий ток. Отличаются друг от друга высотой и типоразмером. Из всех представленных кнопок была выбрана кнопка KLS7-TS6601-4.3-180. Она имеет небольшие размеры, а значит будет занимать меньше места. Таких кнопок понадобится 12 штук: 10 кнопок с цифрами от 0 до 9, 1 кнопка отправки груза и 1 кнопка сброса. На рисунке 12 изображен внешний вид кнопки [12].



Рисунок 12 – Функциональная кнопка

### 3.1.6 Выбор исполнительного устройства

Шаговый двигатель – привод, в последнее время часто используемый в МК-системах. Основное отличие между ним и всеми остальными типами двигателей состоит в способе вращения. Шаговые приводы вращаются «шагами». Каждый шаг представляет собой часть полного оборота. Эта часть

зависит от механического устройства мотора и от способа управления. Использование шаговых двигателей является одним из самых простых, дешевых и легких решений для работы систем точного позиционирования. Поэтому эти двигатели очень часто используются в станках с ЧПУ и в качестве регуляторов [13];

Шаговый двигатель наиболее прост в использовании: при подаче потенциалов на обмотки двигатель повернется строго на определенный угол. Низкая стоимость, по сравнению с остальными двигателями, использование шаговых двигателей является одним из самых простых, дешевых и легких решений для работы систем точного позиционирования. В таблице 4 представлены характеристики некоторых шаговых двигателей.

Таблица 4 – Характеристики шаговых двигателей

Название	Выводы мотора	Шаг	Напряжение, В	Крутящий момент, кгхсм	Ток фазы, мА
42STH60-1206A	6	1,8°	7,2	не менее 6,5	1200
28STH45-0674A	4	1,8°	4,4	не менее 0,9	670
42STH47-0406A	6	1,8°	12	не менее 3,17	400
36HT20-0504MA	4	0,9°	5	не менее 0,95	500

Из представленных шаговых двигателей в таблице 4, двигатель 36HT20-0504MA имеет самый точный шаг – 0,9°, что позволяет разбить один полный оборот на 400 шагов. Биполярный двухфазный шаговый двигатель 36HT20-0504MA – привод, который способен поворачиваться на заданное количество шагов. Выводы мотора – 4 провода: по два на каждую из обмоток, имеющих двух фаз. Рекомендованное напряжение питания мотора 5В. Мощность 5Вт. При этом ток через обмотки составит 500мА. Если устройству сложно получить указанный режим питания, то можно вращать мотор и с помощью меньшего напряжения. В этом случае соответственно снизится потребляемый ток и крутящий момент [13]. На рисунке 13 представлен внешний вид шагового двигателя 36HT20-0504MA.



Рисунок 13 – Шаговый двигатель 36НТ20-0504МА

Для управления двигателями ИУ1 и ИУ2 необходимо устройство, которое бы преобразовывало управляющие сигналы малой мощности в токи, достаточные для управления моторами. Такое устройство называют драйвером двигателей [14].

Микросхема L293D – это сборка из четырёх Н-полумостов. Это может быть использовано для независимого управления двумя двигателями постоянного тока, с возможностью включения реверса. Также чип может быть использован для контроля одного биполярного шагового двигателя [14].

Управление осуществляется через транзисторно-транзисторную логику (TTL), а возвратные (flyback) диоды уже встроены в чип. Это означает, что для управления моторами понадобятся лишь сами моторы, L293D и микроконтроллер. L293D может выдавать до 600 мА на канал. Если этого недостаточно можно соединить несколько микросхем параллельно и тем самым поднять предельные характеристики [14].

Характеристики L293D:

- Напряжение питания управления ( $V_{cc1}$ ): 4,5 – 36 В;
- Напряжение питания нагрузки ( $V_{cc2}$ ):  $V_{cc1}$  – 36 В;
- Максимальный постоянный ток: 600 мА на канал;
- Максимальный пиковый ток (до 100 мкс): 1200 мА на канал [30].

На рисунке 14 представлен внешний вид драйвер L293D.



Рисунок 14 – Драйвер L293D

### 3.1.7 Выбор источника питания

Поскольку использование прототипа системы подразумевает его внедрение в систему автомобиля, то в качестве источника питания системы может послужить аккумулятор автомобиля. Средние показатели напряжения, подаваемого автомобильными аккумуляторами при рабочем состоянии варьируются от 12,5 до 14,5В [2].

Для работы элементов системы требуется напряжение питания равное 5В, соответственно в устройство прототипа системы необходимо включить понижающий преобразователь напряжения, который поддерживает преобразование от заданного диапазона напряжения аккумулятора автомобиля до напряжения в 5В, необходимого для питания системы.

В качестве понижающего преобразователя в прототипе использован DC-DC преобразователь LM2596, работающий от входного напряжения 4-40В и имеющий возможность регулировки напряжения на выходе в пределах от 1,3 до 30В, с максимальным выходным током 3А [17]

На рисунке 15 изображен внешний вид DC-DC преобразователя LM2596.



Рисунок 15 – Преобразователь LM2596

### 3.2 Разработка функциональной схемы прототипа

После того как определены компоненты системы, построим функциональную схему системы круиз-контроля автомобиля, представленную на рисунке 16.

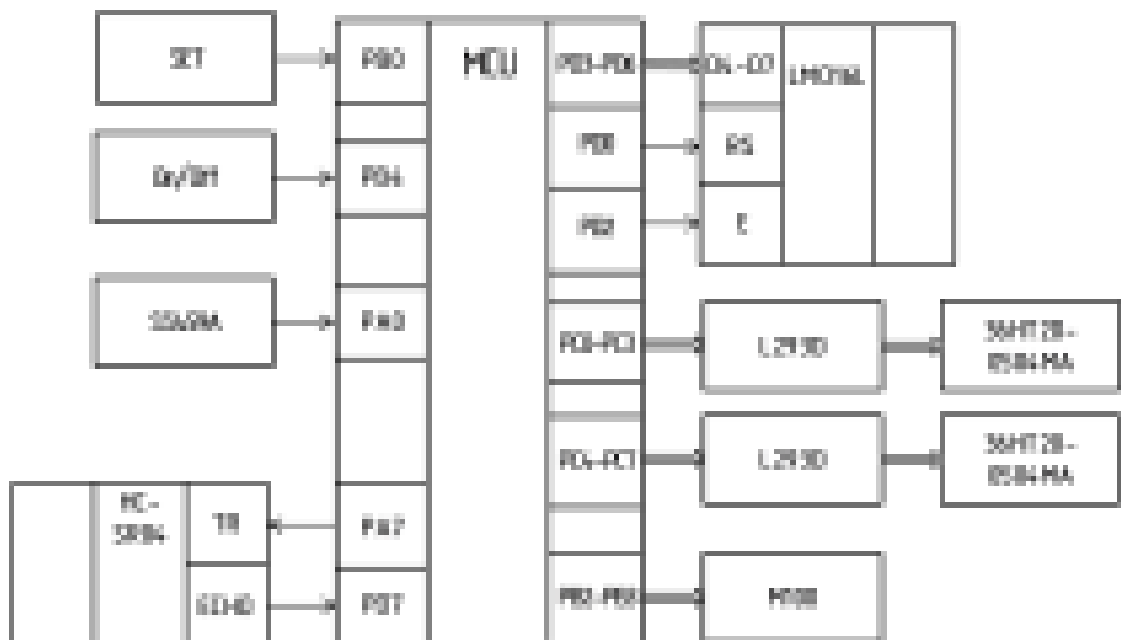


Рисунок 16 – Функциональная система круиз-контроля автомобиля



### **Выводы к разделу 3**

В результате раздела 3, обусловлен выбор компонентов для реализации прототипа системы, выполнено построение функциональной схемы прототипа системы, которая будет являться основой при разработке принципиальной схемы прототипа системы.

#### 4 Разработка принципиальной схемы прототипа системы

В силу того, что на этапе анализа компонентов систем в разделе 3, выявлена невозможность приобретения некоторых компонентов системы и демонстрации их работоспособности в составе модели, прототипирование будет проходить на основе компонентов, более подходящих для тестирования в далее выбранной среде моделирования.

Прототип системы круиз-контроля состоит из следующих элементов:

- 1 Микроконтроллер ATmega8535 (DD1);
- 2 DC-DC преобразователь LM2596 (DD2);
- 3 Два драйвера шагового двигателя L293D (DD3, DD4);
- 4 Четыре конденсатора;
- 5 Светодиод;
- 6 Кварцевый резонатор;
- 7 Катушка индуктивности;
- 8 Шесть резисторов;
- 9 Диод Шоттки;
- 10 Разъёмы.

Микроконтроллер ATmega8535 является «ядром» всей системы. К его портам подключаются светодиоды, резисторы, а также все периферийные устройства системы такие как:

- 1 Драйверы шаговых двигателей L293D – микросхемы DD3, DD4;
- 2 Цифровой дисплей LM016L – разъем XS1
- 3 Звуковые индикаторы – разъемы XS2, XS3;
- 4 Датчик скорости SS459A – разъем XS4;
- 5 Датчик расстояния HC-SR04 – разъем XS5;
- 6 Шаговые двигатели – XS6, XS7;
- 7 Светодиод – HL1;
- 8 Кнопки «SET» и «On/Off» – SW1-SW2.

В таблице 6 представлено подключение разъема XS1 к портам микросхемы DD1.

Таблица 6 – Подключение XS1 к DD1

Входы XS1	Порты DD1	Назначение входов XS1
D4-D7	PD3-PD6	Передача данных
E	PD2	Стrobe-эффект (1-0)
RS	PD0	RS=1 – данные, RS=0 – команда

Входы OUT разъемов XS2 и XS3 подключаются к портам PB2 и PB3 соответственно микросхемы DD1.

Вход OUT разъема XS4 подключается к порту PA0 микросхемы DD1.

Вход ECHO разъема XS5 подключается к порту PD7 микросхемы DD1. Порт PD7 отвечает за внешнее прерывание от порта ECHO. Оно вызывается каждый раз, когда порт PD7 меняет свое логическое состояние.

Вход TR подключается к порту PA7. Порт PA7 подает команду на вход TR разъема XS5.

Для управления шаговыми двигателями XS6, XS7 входы IN1-IN4 драйверов DD3 и DD4 подключаются к портам PC0-PC3 и PC4-PC7 соответственно микросхемы DD1.

В таблице 7 представлено подключение кнопок SW1-SW2 к портам PB0, PB4 микросхемы DD1.

Таблица 7 – Подключение кнопок к микросхеме DD1

Имя кнопки	Диод	Порт подключения	Код кнопки на портах PB1-PB0
SW1	VD2	PB0	01
SW2	VD3	PB1	10

Светодиод HL1 подключен последовательно с кнопкой SW2, соответственно его индикация происходит после получения сигнала через SW2 на порт PB1 от источника питания

На DC-DC преобразователь подается +12В и на выходе напряжение составляет +5В, которое подается на все остальные разъемы. Для работоспособности к преобразователю установлены конденсаторы С1 и С2, диод Шоттки VD1 и катушка индуктивности L1. Катушка индуктивности обеспечивает необходимый номинал выходного напряжения.

#### **Выводы к разделу 4**

В результате раздела 4 выбраны элементы электрической принципиальной схемы прототипа системы, произведено описание подключений устройств в разрабатываемой принципиальной схеме прототипа.



На рисунке 18 представлена граф-схема алгоритма инициализации прототипа системы.



Рисунок 18 – Граф-схема проверки аварийной ситуации

На рисунке 19 представлена граф-схема алгоритма инициализации дисплея



Рисунок 19 – Граф-схема алгоритма инициализации дисплея

На рисунке 20 представлена граф-схема алгоритма поддержания скорости.

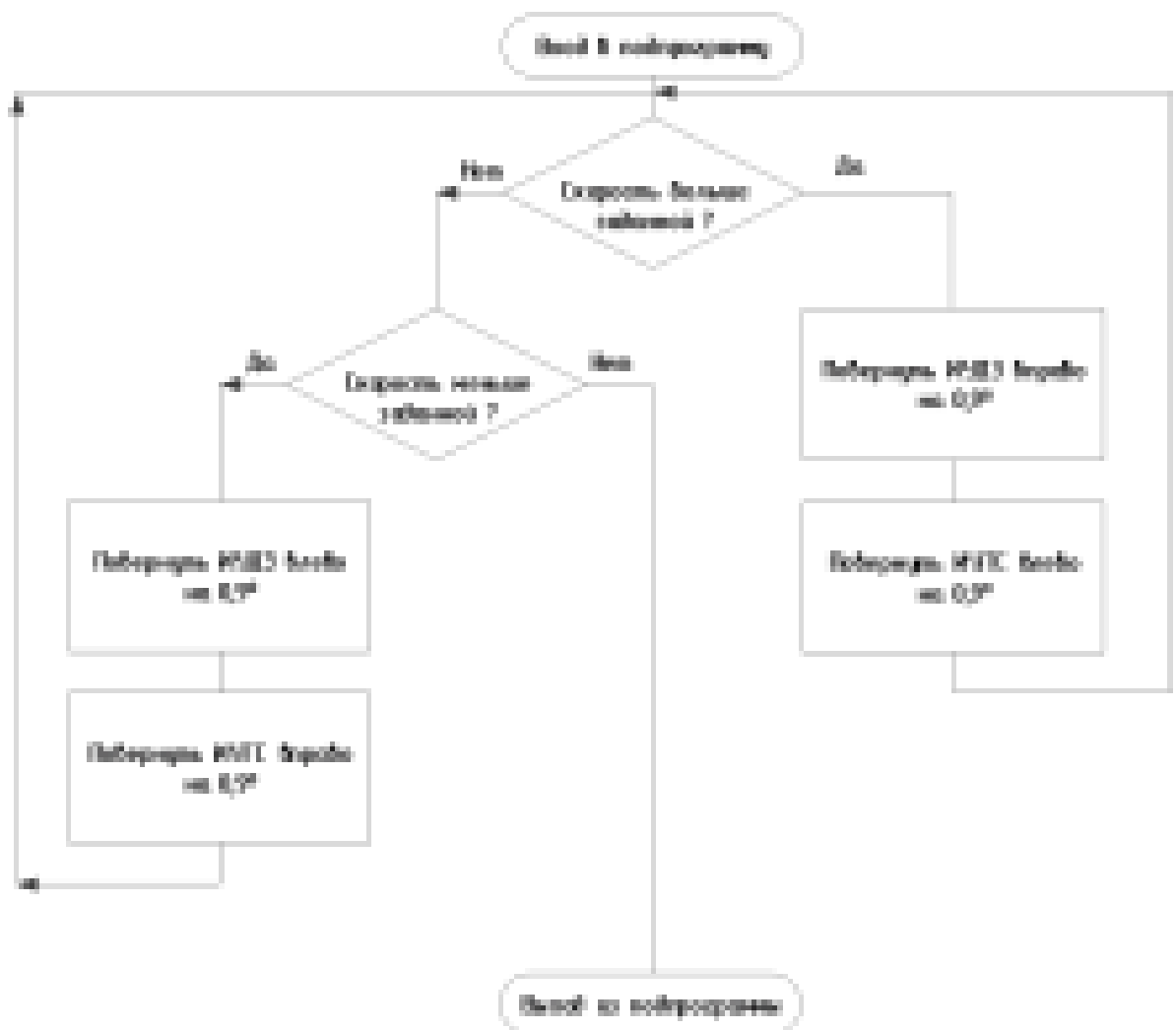


Рисунок 20 – Граф-схема алгоритма поддержания скорости



На рисунке 21 представлена граф-схема алгоритма оповещения об аварийно-опасной ситуации.

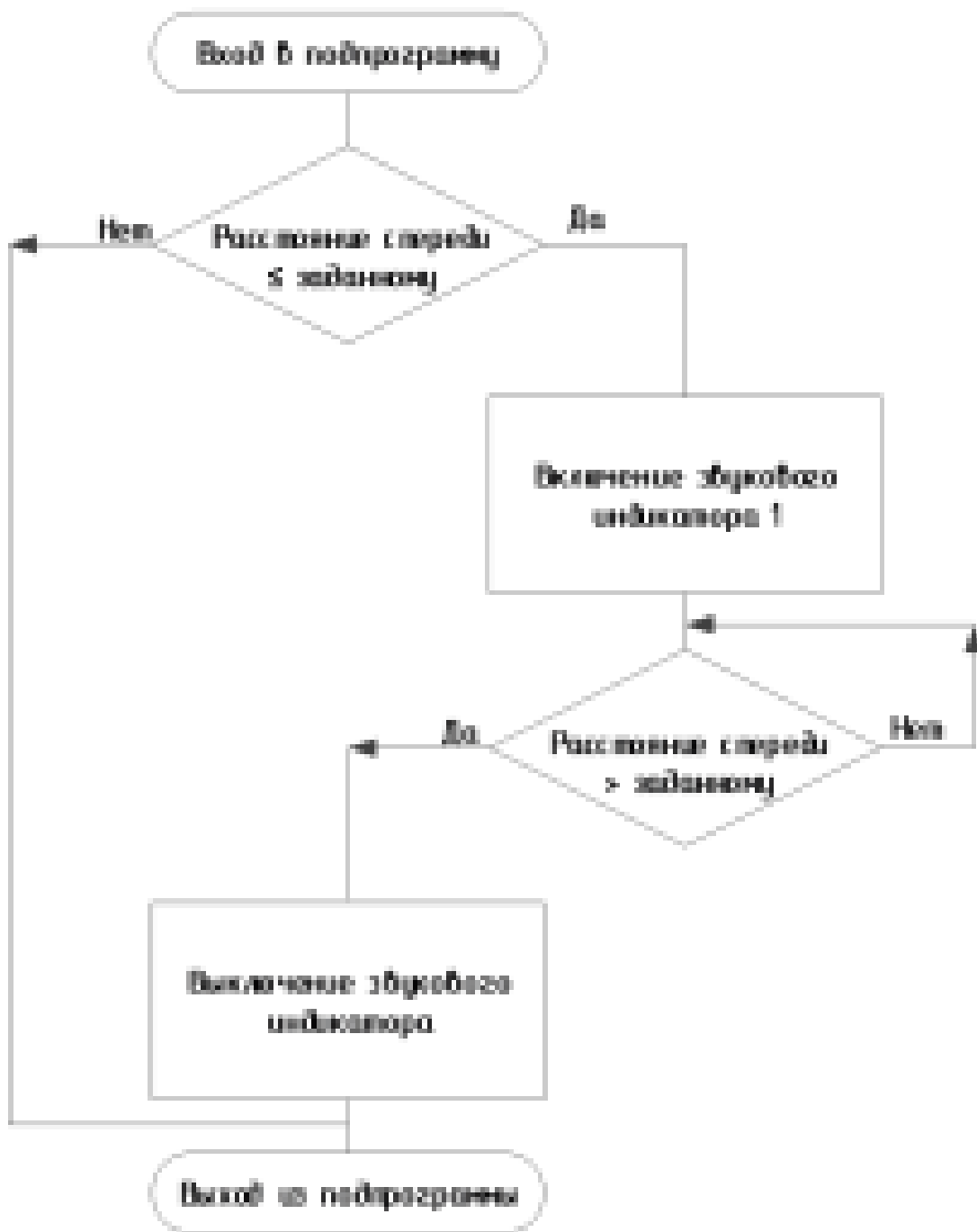


Рисунок 21 – Граф-схема алгоритма поддержания скорости

## 5.2 Обзор средств разработки и тестирования ПО

### 5.2.1 AVR Studio 4

Среда разработки для написания и отладки прикладных программ для микропроцессоров AVR. В данной среде поддерживается большое количество микроконтроллеров фирмы Atmel. Для каждого микроконтроллера используется определенная разновидность языка Assembler, но их мнемоника достаточно схожа. Для работы с МК необходимо изначально его проинициализировать, а затем настроить порты ввода и вывода, а также входы периферийных устройств [18]. Общий вид главного окна программы приведен на рисунке 22.

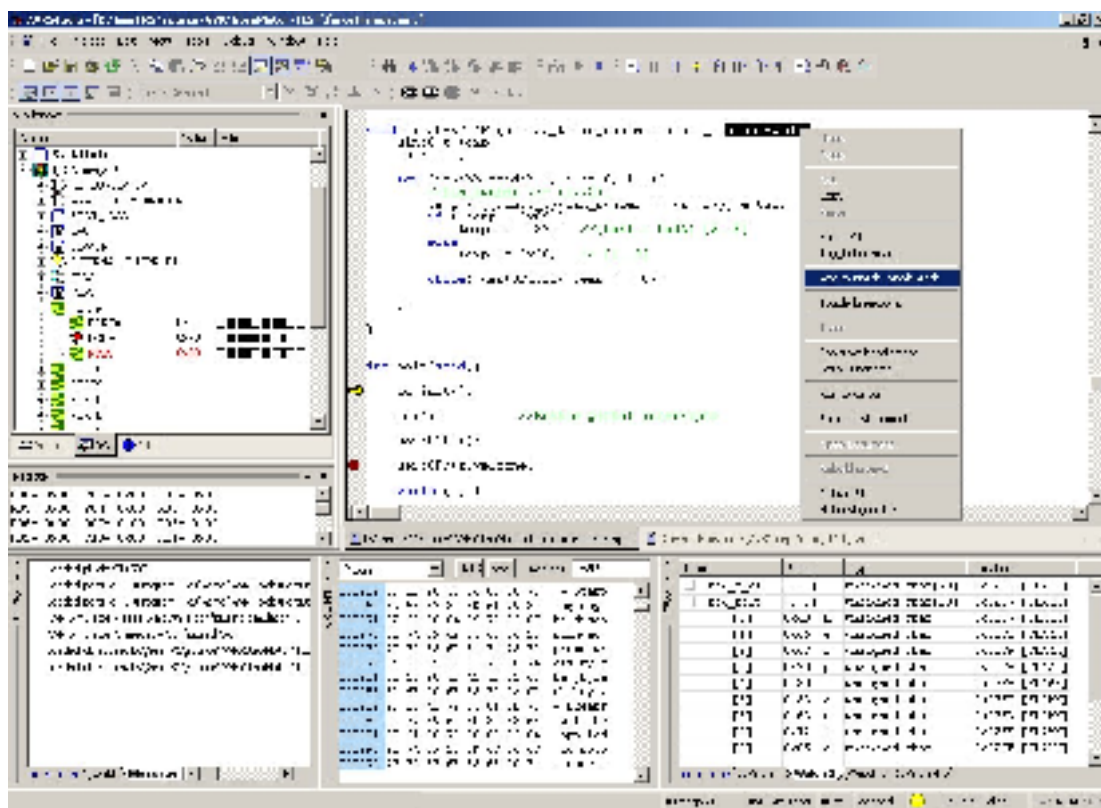


Рисунок 22 – Интерфейс среды разработки ПО AVRStudio 4

AVRStudio поддерживает несколько языков программирования, например: C/C++, Pascal, Delphi, поддерживает Assembler и имеет симулятор для него. Используется в операционных системах Windows 9x/NT/2000/XP/7/8 и выше.

## 5.2.2 Code Vision AVR

Среда разработки для написания и отладки прикладных программ для микропроцессоров AVR. Общий вид главного окна программы приведен на рисунке 23.

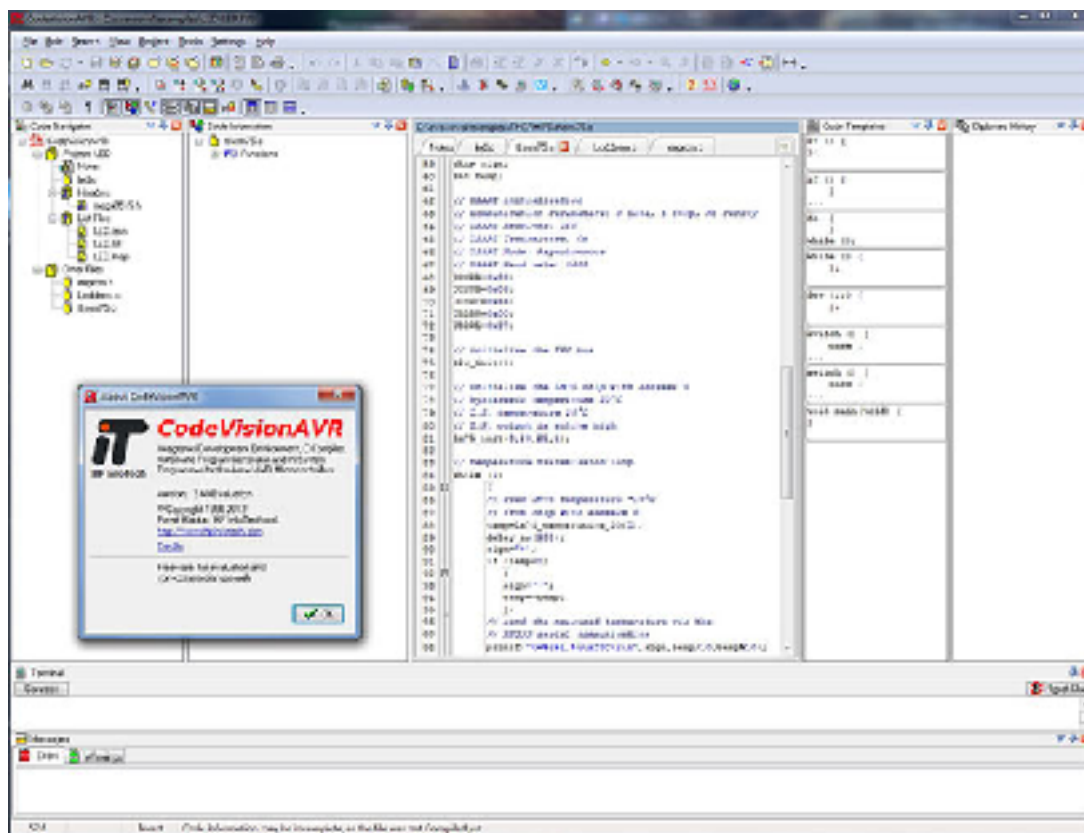


Рисунок 23 – Среда разработки CodeVisionAVR

Работает в операционных системах семейства Windows. Включает в себя: компиляторы C и Assembler, генератор и редактор кода, а также модули взаимодействия с отладочной платой STK-500 [19].

## Выводы к разделу 5

В результате раздела построены граф-схемы для алгоритма прототипа системы круиз-контроля и проведен обзор средств разработки ПО.

## 6 Тестирование аппаратно-программной части прототипа системы в выбранной среде моделирования

### 6.1 Обзор средств моделирования

#### 6.1.1 Proteus

Система автоматизированного проектирования, позволяющая моделировать архитектуру электронных устройств. Программа состоит из нескольких модулей: ISIS и ARES. ISIS предназначен для редактирования электронных схем и имитации их работы, а ARES предназначен для редактирования печатных плат. Кроме этого данное программное обеспечение оснащено встроенным редактором библиотек и автоматической системой размещения элементов на плате. Доступна возможность создавать трехмерные модели печатных плат [20]. Общий вид главного окна программы приведен на рисунке 24.

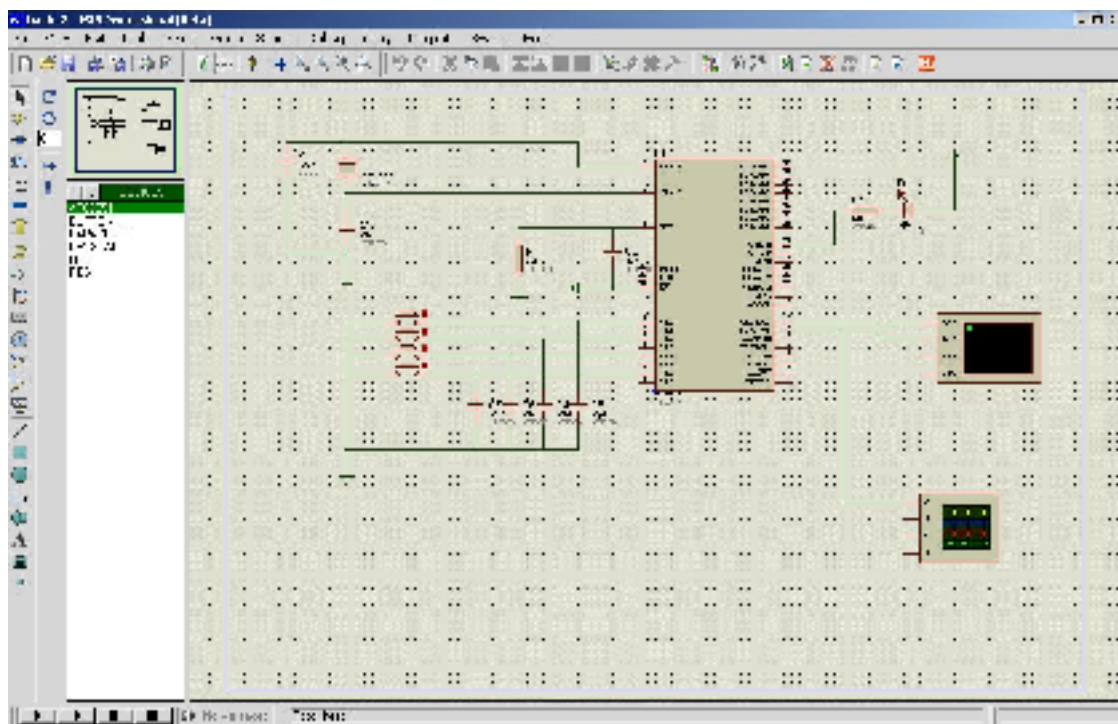


Рисунок 24 – Интерфейс среды моделирования Proteus

В программе располагаются инструменты, позволяющие подключать виртуальные устройства к USB и COM портам компьютера. В последующем, при подключении к этим портам внешних устройств, виртуальная схема работает с ними.

Proteus поддерживает несколько компиляторов: CodeVisionAVR, ICC, WinAVR. Для разработки доступны такие языки программирования как Assembler и C++. Используется в операционных системах Windows 2000/XP/Vista/7 и выше.

### 6.1.2 NI Multisim

Программный пакет для моделирования электронных схем и разводки печатных плат. Имеет очень простой интерфейс и мощные средства для анализа. Встроены виртуальные измерительные приборы с библиотеками реальных устройств. Взаимодействует со средой разработки систем измерения LabVIEW, что помогает сопоставлять теоретические данные с реальными во время создания печатных плат [21]. Общий вид главного окна программы приведен на рисунке 25.

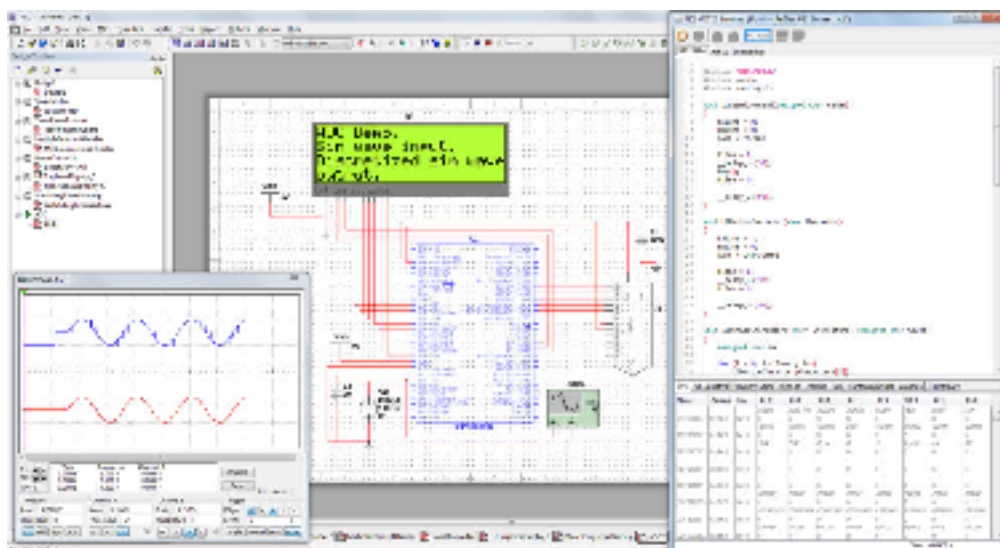


Рисунок 25 – Среда моделирования Multisim

Multisim поддерживает несколько языков программирования, например, Assembler и C++. Может использоваться в операционных системах 32 и 64 разрядов Windows XP/Vista/7 и выше.

## **6.2 Тестирование аппаратно-программного решения**

Из рассмотренных программ для моделирования микроконтроллерной системы электрических схем была выбрана программа автоматизированного проектирования Proteus. Она обладает понятным интерфейсом, большим выбором электронных компонентов, а самое главное, данный программный продукт предоставляет встроенную среду разработки VSM Studio, позволяющую быстро написать программу для микроконтроллера, используемого в проекте, и скомпилировать, не применяя для этого стороннее программное обеспечение.

Написание кода для моделирования осуществлялось на языке C++, т.к. Proteus позволяет использовать данный язык программирования. Язык программирования C++ предоставляет пользователю простую для понимания программную среду, обеспечивая экономию времени при изучении и построении программ для разрабатываемых МК-систем. Код программы моделирования прототипа системы приведен в приложении А.

На рисунке 26 представлен скриншот моделирования системы в программе автоматизированного проектирования Proteus.

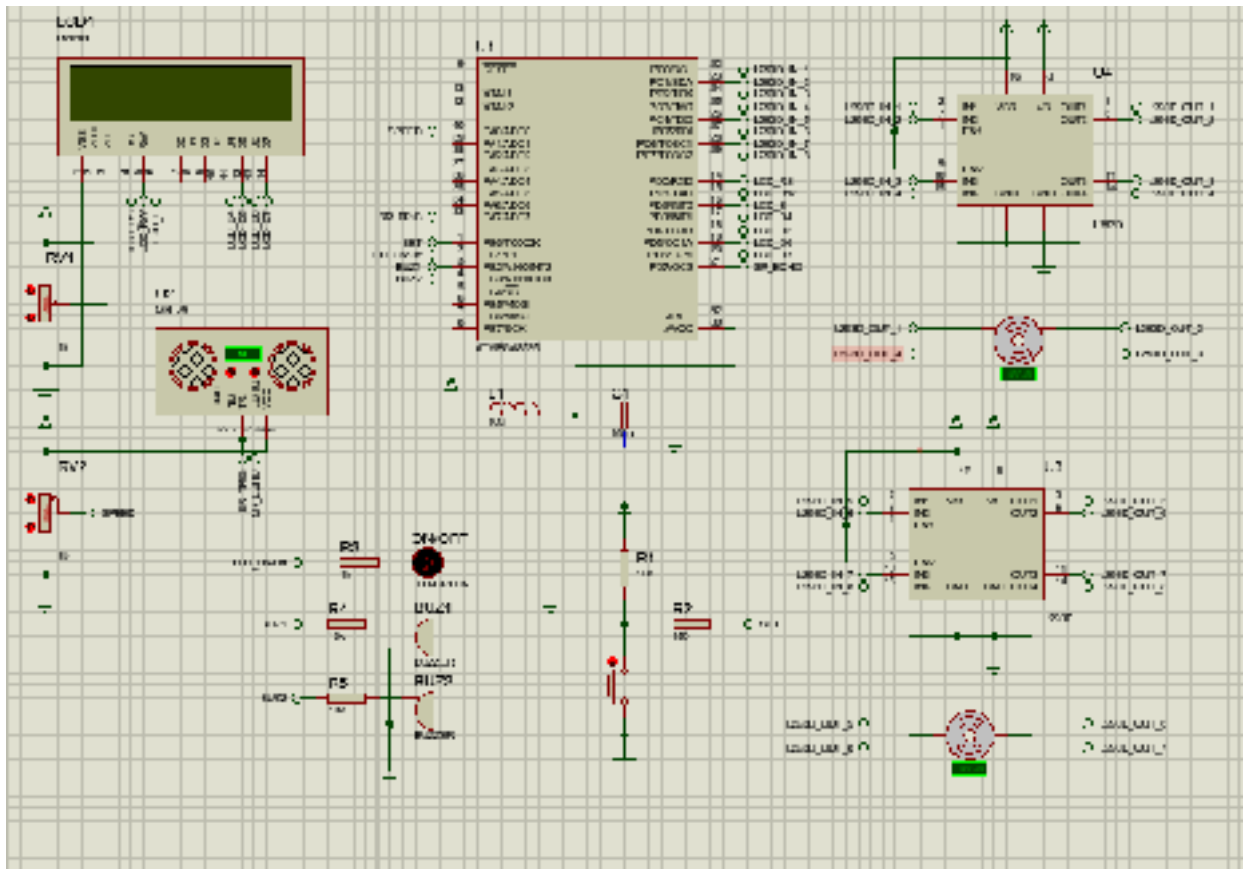


Рисунок 26 – Моделирование системы в Proteus

На рисунке 26 изображено:

- Микроконтроллер ATMEGA8535 (U1);
- Датчики расстояния SRF04 (U2);
- Датчик скорости (RV2);
- Драйверы шагового двигателя L293D (U3,U4);
- Шаговые двигатели;
- Цифровой дисплей (LCD1);
- Функциональная клавиша: «SET».



## **Выводы к разделу 6**

Проведено тестирование аппаратно-программной части прототипа микроконтроллерной системы в среде моделирования Proteus Demo. В ходе моделирования продемонстрирована работоспособность функционала прототипа системы круиз-контроля, заданного в задании на ВКР.

## ЗАКЛЮЧЕНИЕ

В результате выполнения квалификационной работы разработан прототип микроконтроллерной системы круиз-контроля с функциями экстренного торможения, контроля критического расстояния до препятствия и оповещении в случае аварийно-опасной ситуации. В пояснительной записке проведен анализ аналогов разрабатываемой системы, определен программно-аппаратный инструментарий для разработки системы, а также описаны этапы разработки.

Следует так же заметить, что выбор большинства компонентов для реализации прототипа, произведен на основе таких условий как: возможность тестирования в выбранной среде моделирования, сравнительная экономичность, совместимость с остальными компонентами системы.

В результате программно-аппаратного тестирования, прототип системы круиз-контроля автомобиля демонстрирует выполнение всего функционала согласно задания на ВКР.

## СПИСОК СОКРАЩЕНИЙ

- БУ – Блок управления
- ОП – Остановочный путь
- ТЖ – Тормозная жидкость
- ТСА – Тормозная система автомобиля
- ЭСА – Эталонная скорость автомобиля
- ДСА – Действующая скорость автомобиля
- ИУТС – Исполнительное устройство тормозной системы
- ИУДЗ – Исполнительное устройство дроссельной заслонки
- ТВС – Топливо-воздушная смесь
- АЦП – Аналого-цифровой преобразователь
- ВКР – Выпускная Квалификационная Работа
- ДПДЗ – Датчик положения дроссельной заслонки
- МК – Микроконтроллер
- ОС – Операционная система
- ПО – Программное обеспечение
- ШИМ – Широтно-импульсная модуляция
- ACC – Adaptive (Active) Cruise Control
- DSC – Dynamic Stabilization Control
- DIP – Dual in-line Package
- UART – Universal Asynchronous Receiver-Transmitter

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Сайт информационного издания 1gai.ru. Круиз-контроль и адаптивный круиз-контроль: Полное руководство. – Режим доступа: <http://www.1gai.ru/publ/517990-kruiz-kontrol-i-adaptivnyu-kruiz-kontrol-polnoe-rukovodstvo.html>.
- 2 Соснин, Д. А. Электрооборудование и системы бортовой автоматики современных легковых автомобилей: учебное пособие /Д. А. Соснин. – Москва : Солон-Р, 2001. – 272.
- 3 Сайт информационного издания fastmb.ru. Адаптивный круиз-контроль, принцип работы. – Режим доступа: [https://fastmb.ru/auto\\_shem/949-adaptivnyu-kruiz-kontrol-princip-raboty.html](https://fastmb.ru/auto_shem/949-adaptivnyu-kruiz-kontrol-princip-raboty.html).
- 4 Сайт технической и справочной информации по автомобилям BMW: – Режим доступа: <http://tis.bmwcats.com>; – Активный круиз контроль: <http://tis.bmwcats.com/doc1100858/>; – Датчик АСС с блоком управления: <http://tis.bmwcats.com/doc1100853/>.
- 5 Audi Technology Portal. [Электронный ресурс]: – Режим доступа: <https://www.audi-technology-portal.de/en/>; – Adaptive Cruise Control with stop & go function: <https://www.audi-technology-portal.de/en/electrics-electronics/driver-assistant-systems/adaptive-cruise-control-with-stop-go-function>.
- 6 TechAutoPort.ru. [Электронный ресурс]: Системы экстренного торможения автомобиля: устройство, виды и принцип работы. Режим доступа: <https://techautoport.ru/hodovaya-chast/tormoznaya-sistema/sistema-ekstrennogo-tormozheniya.html>.
- 7 Atmel.AVR-ATmega8535. Полное техническая спецификация. - -Режим доступа: <http://ww1.microchip.com/downloads/en/DeviceDoc/doc2502.pdf>.
- 8 Вебинар «mmWave – технология искусственного «зрения» для промышленных и охранных систем» (23.05.18.) [Электронный ресурс] : – Режим доступа: <https://www.compel.ru/2018/04/17/vebinar-mmwave-tehnologiya-iskusstvennogo-zreniya-dlya-promyishlennyih-i-ohrannyih-sistem-23-05-2018>.
- 9 ООО «КОМПЭЛ». Компоненты и модули для производителей электронной аппаратуры. [Электронный ресурс] : – Режим доступа: <https://www.compel.ru/infosheet/ТИ/IWR1443BOOST>.

10 ООО «КОМПЭЛ». Микроволновые радары 76-81ГГц от Texas Instruments. [Электронный ресурс] : – Режим доступа: <https://www.compel.ru/2017/11/17/radar-76-81-ggts-na-odnoy-mikrosheme-s-dalnostyu-do-100-metrov>.

11 ELEC Freaks, Ultrasonic Ranging Module HC – SR04 3стр. // datasheet.

12 Easyelectronics. Обучающие курсы по микроконтроллерам семейств AVR и C51. [Электронный ресурс] : – Режим доступа: <http://easyelectronics.ru/datchik-xolla.html>.

13 AvtoPodium. Магазин автоэлектроники. Звуковые сигналы [Электронный ресурс] : – Режим доступа: <https://avtopodium.net/catalog/stroboskopy-signalny-sgu/signalny-avtomobilnye>.

14 ЧИП И ДИП. Электронные компоненты для приборов. Тактовые кнопки. [Электронный ресурс] : – Режим доступа: <https://www.chipdip.ru/catalog-show/tact-switches?x.153=jeu&x.852=pkM&sort=hits>.

15 NanoJam. 7 популярных приводов. [Электронный ресурс] : – Режим доступа: [https://nanojam.ru/news/7\\_populyarnih\\_privodov\\_dlya\\_robotov](https://nanojam.ru/news/7_populyarnih_privodov_dlya_robotov).

16 Микроконтроллеры. Драйвер двигателей L293D [Электронный ресурс] : – Режим доступа: [https://myrobot.ru/stepbystep/el\\_driver.php](https://myrobot.ru/stepbystep/el_driver.php).

17 ЭЛЕКТРОСАМ.РУ Электрика и электрооборудование, электротехника и электроника – информация! Аккумуляторные батареи. Виды, особенности, устройство, применение [Электронный ресурс] : – Режим доступа: <https://electrosam.ru/glavnaja/jelektrooborudovanie/jelektropitanie/akkumulyatornye-batarei/>.

18 1000VA Интернет-магазин Аккумуляторов и ИБП на рынке с 2004г. Аккумуляторы, аккумуляторные батареи [Электронный ресурс] : – Режим доступа: [http://www.1000va.ru/shop/akkumulyatornie\\_batarei/](http://www.1000va.ru/shop/akkumulyatornie_batarei/).

19 Texas Instruments LM2596 SIMPLE SWITCHER Power Converter 150-kHz 3-A Step-Down Voltage Regulator, Dellsas, Texas, Novemver 1999-Revised May 2018, 45стр. // datasheet.

20 AVRStudio 4 [Электронный ресурс] : AVR Lubic Reference Manual. – Режим доступа: <http://easyelectronics.ru/category/avr-uchebnyj-kurs>.

21 CodeVisionAVR [Электронный ресурс] : – Режим доступа: <http://cxem.net/software/codevisionavr.php> .

22 Wikipedia [Электронный ресурс] : Proteus. – Режим доступа: <https://ru.wikipedia.org/wiki/Proteus>.

23 NI Multisim [Электронный ресурс] : – Режим доступа: <http://cxem.net/software/multisim.php>.

24 ГОСТ 2.743-91. Обозначения условные графические в схемах. Элементы цифровой техники.

25 СТО 4.2-07-2014 "Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности".

## ПРИЛОЖЕНИЕ А

### Код программы моделирования прототипа системы круиз-контроля

#### config.h

```
#ifndef CONFIG_H_
#define CONFIG_H_

#define F_CPU 8000000UL

#define ADC_0 0b01000000;
#define ADC_1 0b01000001;
#define ADC_2 0b01000010;
#define ADC_3 0b01000011;
#define ADC_4 0b01000100;
#define ADC_5 0b01000101;
#define ADC_6 0b01000110;
#define ADC_7 0b01000111;

#define SR_TRIG_PORT PORTC
#define SR_TRIG_DDR DDRC
#define SR_TRIG_PIN 0

#define SR_ECHO_PORT PORTD
#define SR_ECHO_IN PIND
#define SR_ECHO_DDR DDRD
#define SR_ECHO_PIN 2

#define SERVO_PORT PORTC
#define SERVO_DDR DDRC
#define SERVO_PIN_1 1
#define SERVO_PIN_2 2
#define SERVO_PIN_3 3
#define SERVO_PIN_4 4

#define SET_PORT PORTD
#define SET_DDR DDRD
#define SET_IN PIND
#define SET_PIN 4

#define LED_SET_PORT PORTD
```

```
#define LED_SET_DDR DDRD
#define LED_SET_PIN 5

#endif
```

## **main.cpp**

```
#include "config.h"

#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "LCD.h"

float speed, distance;

#pragma region Customize_ADC

void ADC_Init(void)
{
    ADMUX = ADC_0;
    ADCSRA |= (1 << ADPS2)|(1 << ADPS1)|(1 << ADPS0);
    ADCSRA |= (1 << ADEN);
}

float ADC_Read(void)
{
    ADCSRA |= (1 << ADSC);
    while((ADCSRA & (1 << ADIF)) == 0);
    return (float) ADCW * 0.004883;
}

#pragma endregion Customize_ADC

#pragma region Customize_Servo

struct Servo_Pins
{
    int Pin;
    Servo_Pins *Next;
    Servo_Pins *Prev;
};

Servo_Pins S1, S2, S3, S4;
```



```

void Servo_Init()
{
    SERVO_PORT  &=  ~((1  <<  SERVO_PIN_1)|(1  <<  SERVO_PIN_2)|(1  <<
SERVO_PIN_3)|(1 << SERVO_PIN_4));
    SERVO_DDR |= (1 << SERVO_PIN_1)|(1 << SERVO_PIN_2)|(1 << SERVO_PIN_3)|(1 <<
SERVO_PIN_4);

    S1.Pin = SERVO_PIN_1;
    S2.Pin = SERVO_PIN_2;
    S3.Pin = SERVO_PIN_3;
    S4.Pin = SERVO_PIN_4;

    S1.Next = &S2;
    S2.Next = &S3;
    S3.Next = &S4;
    S4.Next = &S1;

    S1.Prev = &S4;
    S4.Prev = &S3;
    S3.Prev = &S2;
    S2.Prev = &S1;

}

#pragma endregion Customize_Servo

#pragma region LCD_Print

void LCD_Print_KPH(float data)
{
    char text[10];
    sprintf(text, "%d kph ", abs(data));
    lcdGotoXY(0, 4);
    _delay_ms(5);
    lcdPuts(text);
}

#pragma endregion LCD_Print

#pragma region Customize_Timer

void Timer2_Init(void)
{
    TCNT2 = 0;
    TIMSK |= (1 << TOIE2);
    TCCR2 |= (1 << CS22)|(1 << CS21)|(1 << CS20);
}

ISR(TIMER2_OVF_vect)
{

```

```

        LCD_Print_KPH(speed);
    }

#pragma endregion Customize_Timer

void Setup(void)
{
    _delay_ms(100);
    lcdInit();
    _delay_ms(5);
    lcdClear();
    _delay_ms(5);
    lcdSetDisplay(LCD_DISPLAY_ON);
    _delay_ms(5);
    lcdSetCursor(LCD_CURSOR_OFF);

    char text[5];
    strcpy(text, "S = ");
    lcdGotoXY(0, 0);
    _delay_ms(5);
    lcdPuts(text);

    SET_PORT |= (1 << SET_PIN);
    SET_DDR &= ~(1 << SET_PIN);

    LED_SET_PORT &= ~(1 << LED_SET_PIN);
    LED_SET_DDR |= (1 << LED_SET_PIN);

    ADC_Init();
    Servo_Init();

    Timer2_Init();
    sei();
}

void Loop(void)
{
    Servo_Pins *current = &S1;

    while(1)
    {
        speed = ADC_Read()*30;
        if (((SET_IN & (1 << SET_PIN)) == 0) && speed >= 40)
        {
            LED_SET_PORT |= (1 << LED_SET_PIN);
            if (speed <= 50)
            {
                SERVO_PORT |= (1 << current->Pin);
                _delay_ms(2000);
                SERVO_PORT &= ~(1 << current->Pin);
            }
        }
    }
}

```

```

        current = current->Next;
    }

    if (speed >= 90)
    {
        current = current->Prev;
        SERVO_PORT |= (1 << current->Pin);
        _delay_ms(2000);
        SERVO_PORT &= ~(1 << current->Pin);
    }
}
else
{
    LED_SET_PORT &= ~(1 << LED_SET_PIN);
}
_delay_ms(500);
}
}

int main(void)
{
    Setup();
    Loop();
}

```

## LCD.cpp

```

#include "config.h"
#include <avr/io.h>
#include <util/delay.h>
#include <avr/pgmspace.h>
#include <avr/eeprom.h>
#include <avr/interrupt.h>
#include "LCD.h"

void lcdSendNibble(char byte, char state)
{
    LCDCONTROLDDR |= 1<<LCD_RS | 1<<LCD_RW | 1<<LCD_E;
    LCDDATADDR |= 1<<LCD_D4 | 1<<LCD_D5 | 1<<LCD_D6 | 1<<LCD_D7;
    LCDCONTROLPORT &= ~(1<<LCD_RW);
    if (state)
    {
        LCDCONTROLPORT |= 1<<LCD_RS;
    }
    else
    {
        LCDCONTROLPORT &= ~(1<<LCD_RS);
    }
    LCDCONTROLPORT |= 1<<LCD_E;
    LCDDATAPORT &= ~(1<<LCD_D4 | 1<<LCD_D5 | 1<<LCD_D6 | 1<<LCD_D7);
}

```

```

    if (byte & (1<<3))
    {
        LCDDATAPORT |= 1<<LCD_D7;
    }
    if (byte & (1<<2))
    {
        LCDDATAPORT |= 1<<LCD_D6;
    }
    if (byte & (1<<1))
    {
        LCDDATAPORT |= 1<<LCD_D5;
    }

    if (byte & (1<<0))
    {
        LCDDATAPORT |= 1<<LCD_D4;
    }

    delay_us(LCD_STROBEDELAY_US);
    LCDCONTROLPORT &= ~(1<<LCD_E);
}

char lcdGetNibble(char state)
{
    char temp = 0;
    LCDCONTROLDDR |= 1<<LCD_RS | 1<<LCD_RW | 1<<LCD_E;
    LCDCONTROLPORT |= 1<<LCD_RW;
    if (state)
    {
        LCDCONTROLPORT |= (1<<LCD_RS);
    }
    else
    {
        LCDCONTROLPORT &= ~(1<<LCD_RS);
    }
    LCDCONTROLPORT |= 1<<LCD_E;

    LCDDATADDR &= ~(1<<LCD_D4 | 1<<LCD_D5 | 1<<LCD_D6 | 1<<LCD_D7);

    LCDDATAPORT |= 1<<LCD_D4 | 1<<LCD_D5 | 1<<LCD_D6 | 1<<LCD_D7;

    _delay_us(LCD_STROBEDELAY_US);

    LCDCONTROLPORT &= ~(1<<LCD_E);

    if (LCDDATAPIN & (1<<LCD_D7))
    {
        temp |= 1<<3;
    }
    if (LCDDATAPIN & (1<<LCD_D6))

```

```

    {
    temp |= 1<<2;
    }

    if (LCDDATAPIN & (1<<LCD_D5))
    {
    temp |= 1<<1;
    }

    if (LCDDATAPIN & (1<<LCD_D4))
    {
    temp |= 1<<0;
    }

    return temp;
}

char lcdRawGetByte(char state)
{
    char temp = 0;

    temp |= lcdGetNibble(state);
    temp = temp<<4;
    temp |= lcdGetNibble(state);

    return temp;
}

void lcdRawSendByte(char byte, char state)
{
    lcdSendNibble((byte>>4), state);
    lcdSendNibble(byte,state);
}

char lcdIsBusy(void)
{
    _delay_ms(DELAY);
    return 0x00;
}

void lcdInit(void)
{
    while (lcdIsBusy()) ;
    lcdSendNibble(0b0010, LCD_COMMAND);
    while (lcdIsBusy()) ;
    lcdRawSendByte(0b00101000, LCD_COMMAND);
    while (lcdIsBusy()) ;
    lcdRawSendByte(0b00000001, LCD_COMMAND);
    while (lcdIsBusy()) ;
    lcdRawSendByte(0b00000110, LCD_COMMAND);
}

```

```

        while (lcdIsBusy());
        lcdRawSendByte(0b00001100, LCD_COMMAND);
    }

void lcdSetCursor(char cursor)
{
    while (lcdIsBusy());

    lcdRawSendByte((0b00001100 | cursor), LCD_COMMAND);
}

void lcdSetDisplay(char state)
{
    while (lcdIsBusy());

    lcdRawSendByte((0b00001000 | state), LCD_COMMAND);
}

void lcdClear(void)
{
    while (lcdIsBusy());

    lcdRawSendByte(0b00000001, LCD_COMMAND);
}

void lcdGotoXY(char str, char col)
{
    while (lcdIsBusy());

    lcdRawSendByte((0b10000000 | ((0x40 * str) + col)), LCD_COMMAND);
}

void lcdDisplayScroll(char pos, char dir)
{
    while (pos)
    {
        while (lcdIsBusy());

        lcdRawSendByte((0b00011000 | dir), LCD_COMMAND);
        pos--;
    }
}

void lcdPuts(char *str)
{
    while (*str)
    {
        while (lcdIsBusy());

        lcdRawSendByte(*str++, LCD_DATA);
    }
}

```

```

    }
}

void lcdPutsf(char *str)
{
    while (pgm_read_byte(str))
    {
        while (lcdIsBusy());

        lcdRawSendByte(pgm_read_byte(str++), LCD_DATA);
    }
}

void lcdPutse(uint8_t *str)
{
    while (eeprom_read_byte(str))
    {
        while (lcdIsBusy());

        lcdRawSendByte((char)(eeprom_read_byte(str++)), LCD_DATA);
    }
}

void lcdLoadCharacter(char code, char *pattern)
{
    while (lcdIsBusy());

    lcdRawSendByte((code<<3) | 0b01000000, LCD_COMMAND);

    for (char i = 0; i <= 7; i++)
    {
        while (lcdIsBusy());

        lcdRawSendByte(*pattern++, LCD_DATA);
    }
    while (lcdIsBusy());
    lcdRawSendByte(0b10000000, LCD_COMMAND);
}

void lcdLoadCharacterf(char code, char *pattern)
{
    while (lcdIsBusy());

    lcdRawSendByte((code<<3) | 0b01000000, LCD_COMMAND);

    for (char i = 0; i <= 7; i++)
    {
        while (lcdIsBusy());

        lcdRawSendByte(pgm_read_byte(pattern++), LCD_DATA);
    }
}

```

```

    }
    while (lcdIsBusy());
    lcdRawSendByte(0b10000000, LCD_COMMAND);
}

```

## LCD.h

```

#ifndef LCD_H_
#define LCD_H_

#define LCDDATAPORT    PORTB
#define LCDDATADDR    DDRB
#define LCDDATAPIN    PINB
#define LCD_D4        3
#define LCD_D5        4
#define LCD_D6        5
#define LCD_D7        6

#define LCDCONTROLPORT PORTB
#define LCDCONTROLDDR  DDRB
#define LCD_RS        0
#define LCD_RW        1
#define LCD_E         2

#define LCD_STROBEDELAY_US 5

#define LCD_COMMAND    0
#define LCD_DATA      1

#define LCD_CURSOR_OFF 0
#define LCD_CURSOR_ON  2
#define LCD_CURSOR_BLINK 3

#define LCD_DISPLAY_OFF 0
#define LCD_DISPLAY_ON  4

#define LCD_SCROLL_LEFT 0
#define LCD_SCROLL_RIGHT 4

#define LCD_STROBDOWN  0
#define LCD_STROBUP    1

#define DELAY          1

void lcdSendNibble(char byte, char state);
char lcdGetNibble(char state);
char lcdRawGetByte(char state);
void lcdRawSendByte(char byte, char state);
char lcdIsBusy(void);
void lcdInit(void);

```



```
void lcdSetCursor(char cursor);
void lcdSetDisplay(char state);
void lcdClear(void);
void lcdGotoXY(char str, char col);
void lcdDisplayScroll(char pos, char dir);
void lcdPuts(char *str);
void lcdPutsf(char *str);
void lcdPutse(uint8_t *str);
void lcdLoadCharacter(char code, char *pattern);
void lcdLoadCharacterf(char code, char *pattern);
void lcdLoadCharacterere(char code, char *pattern);

#endif
```