

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

РАБОТА ПРОВЕРЕНА

Рецензент

\_\_\_\_\_ С. А. Вашляев  
«\_\_» \_\_\_\_\_ 2020 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой ЭВМ

\_\_\_\_\_ Г.И. Радченко  
«\_\_» \_\_\_\_\_ 2020 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Руководитель работы,  
к.т.н., доцент каф. ЭВМ

\_\_\_\_\_ И.Л. Кафтанников  
«\_\_» \_\_\_\_\_ 2020 г.

Автор работы,  
студент группы КЭ-405

\_\_\_\_\_ А.В. Кравчук  
«\_\_» \_\_\_\_\_ 2020 г.

Нормоконтролёр,  
ст. преп. каф. ЭВМ

\_\_\_\_\_ С. В. Сяськов  
«\_\_» \_\_\_\_\_ 2020 г.

Челябинск-2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_ Г.И. Радченко

«\_\_\_» \_\_\_\_\_ 2020 г.

### **ЗАДАНИЕ**

**на выпускную квалификационную работу бакалавра**  
студенту группы КЭ-405  
Кравчуку Андрею Владимировичу  
обучающемуся по направлению  
09.03.01 «Информатика и вычислительная техника»

- 1. Тема работы:** «**Разработка программно-аппаратного комплекса для диагностики автомобиля**» утверждена приказом по университету от 24 апреля 2020 г. №627
- 2. Срок сдачи студентом законченной работы:** 16 июня 2020 г.
- 3. Исходные данные к работе:**
  - среда разработки Keil uVision;
  - микроконтроллер stm32f103c8t6;
  - reference manual stm32f103;
  - programming manual stm32f103;
  - datasheet stm32f103.

**4. Перечень подлежащих разработке вопросов:**

- рассмотрение и анализ существующих приборов, определение списка их достоинств и недостатков;
- изучение методов коммуникации диагностического оборудования с электронными системами автомобильного транспорта;
- выбор компонентов для разработки аппаратной части комплекса;
- выбор инструментария для разработки программной части комплекса;
- разработка программно-аппаратного комплекса;
- проверка работоспособности программно-аппаратного комплекса в реальных условиях.

**5. Дата выдачи задания:** 1 декабря 2019 г.

Руководитель работы \_\_\_\_\_ /И.Л.Кафтанников/

Студент \_\_\_\_\_ /А.В.Кравчук /

## КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	30.12.2019	
Разработка модели, проектирование	01.02.2020	
Реализация системы	15.04.2020	
Тестирование, отладка, эксперименты	10.05.2020	
Компоновка текста работы и сдача на нормоконтроль	15.06.2020	
Подготовка презентации и доклада	16.06.2020	

Руководитель работы \_\_\_\_\_ /И.Л.Кафтанников/

Студент \_\_\_\_\_ /А.В.Кравчук/

## Аннотация

А. В. Кравчук. Разработка программно-аппаратного комплекса для диагностики автомобиля. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2020, 84с., 27 ил., библиогр. список – 34 наим.

В рамках выпускной квалификационной работы производится обзор, анализ, выявление сильных и слабых сторон существующих программно-аппаратных комплексов для диагностики автомобильного транспорта. Осуществляется разработка программно-аппаратного комплекса для диагностики электронных систем автомобиля и наблюдения за параметрами этих систем в режиме реального времени. Подробно рассматриваются основные способы коммуникации диагностического оборудования и электроники транспортных средств. Доказывается актуальность проблем нехватки отечественного оборудования на внутреннем рынке.

# ОГЛАВЛЕНИЕ

ГЛОССАРИЙ.....	8
ВВЕДЕНИЕ.....	9
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	11
1.1. ОБЗОР АНАЛОГОВ .....	12
1.2. АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ .....	15
1.2.1. Коммуникационный протокол SAE J1850.....	16
1.2.2. Коммуникационный протокол KWP-2000.....	17
1.2.3. Коммуникационный протокол CAN2.0A.....	19
1.2.4. Обзор микроконтроллеров .....	22
1.2.5. Обзор инструментария.....	24
1.2.6. Устройства согласования логических уровней .....	26
1.2.7. Стабилизатор напряжения.....	28
1.2.8. Язык программирования микроконтроллера .....	29
1.2.9. Библиотека, используемая для написания программы .....	30
1.3. ВЫВОД .....	32
2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ .....	33
2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ .....	33
2.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ.....	35
2.2.1. Требования, предъявляемые к устройству: .....	35
2.2.2. Требования, предъявляемые к диагностируемому автомобилю: .....	36
2.2.3. Требования, предъявляемые к диагносту: .....	36
3. ПРОЕКТИРОВАНИЕ.....	37
3.1. АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ.....	37
3.2. АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧ .....	39
3.3. ОПИСАНИЕ ДАННЫХ .....	44

4.	РЕАЛИЗАЦИЯ .....	46
4.1.	РЕАЛИЗАЦИЯ ТЕСТОВОГО УСТРОЙСТВА.....	46
5.	ТЕСТИРОВАНИЕ .....	57
5.1.	МЕТОДОЛОГИЯ ТЕСТИРОВАНИЯ.....	57
5.2.	ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ .....	57
6.	ЗАКЛЮЧЕНИЕ .....	62
	БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	63
	ПРИЛОЖЕНИЕ А .....	66
	ПРИЛОЖЕНИЕ Б.....	67
	ПРИЛОЖЕНИЕ В .....	68

## ГЛОССАРИЙ

*ЭСУД* – Электронная система управления двигателем.

*ЭБУ* – электронный блок управления.

*Диагностический тестер* – прибор, предназначенный для чтения ошибок в электронных системах автомобиля, а также взаимодействия с его электронными блоками.

*SAE* – Общество автомобильных инженеров (англ. Society of Automotive Engineers).



## ВВЕДЕНИЕ

Компьютерная автомобильная диагностика - одно из самых популярных и бурно развивающихся направлений в автомобильной промышленности, однако, отечественные разработки в этой области практически отсутствуют ввиду отсутствия в открытом доступе какой-либо информации, касающейся проектирования автомобильного диагностического оборудования.

Наличие качественного диагностического оборудования в автосервисе необходимо по той причине, что грамотно определенная неисправность позволяет владельцу автомобиля сэкономить деньги, а работникам автосервиса не приходится тратить лишнее время на устранение неисправности.

К современным программно-аппаратным комплексам для диагностики выдвигается все большее число требований, это ведет к тому, что функциональность и сложность современного диагностического оборудования со временем значительно увеличивается. Одним из важнейших требований к такому роду оборудования является устойчивость к вибрациям, помехам, наведенным системой зажигания во время диагностики запущенного автомобиля, защита от скачков напряжения в бортовой сети.[1]

С помощью компьютерной диагностики можно произвести следующие операции:

- оценить состояние узлов и агрегатов автомобиля;
- считать и удалить коды неисправностей;
- посмотреть параметры систем автомобиля в реальном времени.

Тема создания оборудования для автомобильной диагностики актуальна и по сей день по различным причинам:

- появление ранее не существовавших систем, облегчающих управление автомобилем в критических ситуациях;

- рост числа электронных блоков и увеличение сложности электрической схемы автомобилей;
- изменение методик управления двигателем (например, добавление новых датчиков для соответствия стандартам);
- отсутствие на рынке качественного недорогого диагностического оборудования.

Цель работы – проектирование и разработка программно-аппаратного комплекса для диагностики автомобиля, с возможностью чтения и удаления кодов неисправностей, а также наблюдения за параметрами электронной системы управления двигателем в реальном времени.

Для того чтобы достигнуть поставленной цели, необходимо решить следующий ряд задач:

- произвести исследование существующих на данный момент приборов;
- изучить каким образом диагностическое оборудование получает данные от электронных блоков автомобиля;
- разработать программно-аппаратный комплекс для диагностики автомобиля и наблюдения за параметрами электронных блоков в режиме реального времени;
- провести испытания разработанного программно-аппаратного комплекса, выявить достоинства и недостатки.

На сегодняшний день разработаны, поддерживаются и обновляются определенные стандарты, из которых можно узнать о применяемых протоколах обмена данными между диагностическим тестером и автомобилем. В текущей работе эти стандарты были полностью изучены, проект разрабатывался в полном соответствии с ними.

Работа состоит из глоссария, введения, пяти разделов, заключения и библиографии. Объем работы составляет 84 страницы, объем библиографии – 34 источника. Введение содержит краткое описание сферы автомобильной диагностики, в нем также доказывается востребованность разработки соответствующего оборудования. Глава «Анализ предметной области» посвящена обзору существующих приборов для диагностики автомобиля, анализу их достоинств и недостатков, и применяемых в ходе работы технологических решений. Глава «Определение требований» содержит информацию об основных функциях системы, приводится схема взаимодействия ее объектов. Приводятся списки функциональных и нефункциональных требований. В главе «Проектирование» уделено особое внимание проектированию архитектуры системы, приводятся алгоритмы решения поставленных задач, описание данных, отправляемых диагностическим тестером электронным блокам автомобиля. В главе «Реализация» приводится подробное описание компонентов системы, представлен макет, собранный для тестирования программно-аппаратного комплекса. Глава «Тестирование» содержит информацию о тестовом запуске системы. В заключении приводятся основные выводы о проделанной работе, подводятся итоги этапов проектирования, реализации, тестирования программно-аппаратного комплекса.

## **1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

История электронной автомобильной диагностики начинается в США в 80-ых годах, когда крупнейшая американская автомобильная корпорация представила интерфейс ALDL и необходимый протокол для тестирования модулей управления двигателем. Эта система была нечетко стандартизирована и страдала от того, что технические характеристики канала связи варьировались

от одной модели к другой. Интерфейс ALDL в основном использовался производителями для диагностики в их дилерских и официальных сервисных центрах. Существовало 3 варианта коннекторов для соединения со сканером, кроме того, назначение и количество контактов в каждом варианте коннектора было своим.

В 1988 году Совет по воздушным ресурсам штата Калифорния обязал наделить все продаваемые автомобили системами бортовой диагностики. Расположение диагностического разъема, его тип, а также диагностический протокол не были определены каким-либо стандартом.

В 1996 году был принят стандарт OBD-II, использующийся до сих пор. Начиная с 2004 года, все выпускаемые автомобили должны были оснащаться бортовыми системами диагностики. При возникновении неисправности, на приборной панели автомобиля загорается лампа MIL (т.н. лампа «Check Engine»). В данном случае необходимо прочесть ошибку специальным диагностическим оборудованием, а после – стереть ее из памяти электронных блоков управления [2].

## **1.1. ОБЗОР АНАЛОГОВ**

На сегодняшний день на рынке присутствует в продаже множество различных диагностических сканеров для автомобиля. Они во многом помогают специалистам обнаружить неисправность и устранить ее в кратчайшие сроки. Однако если сканер не может считать код ошибки и/или отобразить некоторые параметры систем, поиск требующей замены части может затянуться не на один час, а в случае неверного определения мастером неисправности, автосервис может понести значительные материальные убытки.

Для определения достоинств и недостатков были выбраны три наиболее часто используемых диагностических сканера в профессиональных автосервисах: сканер диагностический Launch Pilot Scan, мультимарочный диагностический сканер Autel Maxi. Sys MS906, сканер диагностический Launch X431 Pro. Рассмотрим подробно каждый из них.

#### 1.1.1 Сканер диагностический Launch Pilot Scan [3]

Достоинства:

- небольшие габариты;
- информативность выводимой информации;
- большой функционал;
- автоматическая идентификация автомобилей;

Недостатки:

- невозможность работы при разряженной батарее;
- высокая стоимость;
- ограниченное число автомобилей в диагностической базе;
- невозможность выбора диагностического протокола;
- невозможность отправки произвольной команды по CAN-шине;
- невозможность отправки произвольной команды по K-линии.

#### 1.1.2 Мультимарочный диагностический сканер Autel Maxi. Sys MS906 [4]

Достоинства:

- автоматическая идентификация блоков управления;
- сервисные функции;
- функции кодирования;
- обновления в один клик;

- возможность записи логов диагностики.

Недостатки:

- высокая стоимость;
- платные обновления;
- некорректная работа ПО после некоторых обновлений;
- низкое качество техподдержки;
- низкое качество комплектующих;
- невозможность выбора диагностического протокола;
- невозможность отправки произвольной команды по CAN-шине;
- Невозможность отправки производной команды по K-линии.

### 1.1.3 Сканер диагностический Launch X431 Pro [5]

Достоинства:

- коммуникация с автомобилем с помощью технологии Bluetooth;
- большой функционал;
- информативность выводимой информации.

Недостатки:

- нестабильная работа после некоторых обновлений;
- невозможность отправки произвольной команды по CAN-шине;
- невозможность отправки производной команды по K-линии;
- невозможность работы при разряженном аккумуляторе.

Все вышеперечисленные приборы имеют схожий функционал, но всех их объединяет невозможность отправки команды по CAN-шине и K-линии, а также высокая стоимость [6]. Также на некоторых из этих приборов, часто

невозможна работа вблизи системы зажигания автомобиля, когда на соединительные провода действуют помехи [7].

Сведем результаты в таблицу.

Таблица 1.1 – Преимущества и недостатки аналогов

Критерий	Launch Pilot Scan	Autel MS906	Launch X431 Pro
Возможность ручного выбора линии связи	-	-	+
Зависимость от батареи	+	+	+
Возможность отправки произвольной команды	-	-	-
Качество исполнения	+	-	+
Стоимость, руб	26000	120000	От 65000

## 1.2. АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ

В данном разделе рассматриваются основные технологические решения, примененные при проектировании программно-аппаратного комплекса.

Далее рассматриваются 3 диагностических протокола: SAE J1850, KWP2000 и CAN2.0A. На сегодняшний день, прибор, обладающий возможностью отправки и приема запроса по этим протоколам способен определить неисправности у большинства автомобилей.

### 1.2.1. Коммуникационный протокол SAE J1850

В настоящее время существуют две разновидности протокола SAE J1850 для двух линий связи: J1850 VPW и J1850 PWM. Главное отличие между ними состоит в скорости передачи данных и способе кодирования данных.

SAE J1850 VPW является низкоскоростным (максимальная скорость обмена данными равна 10,4 кбайт/с). Для передачи данных используется один провод. Максимальная длина линии для передачи данных равна 40м, а максимально возможное количество подключенных устройств – 32. Тип кодирования – длительность импульса [15].

SAE J1850 PWM является высокоскоростным (максимальная скорость обмена данными равна 41,6 кбайт/с). Для передачи данных используется два провода. Сигнал является дифференциальным. Максимальная длина линии для передачи данных равна 40м, а максимально возможное количество подключенных устройств – 32. Тип кодирования – широтно-импульсная модуляция [16].

Формат кадра, передаваемого по протоколу SAE J1850, приведен на рисунке 1.2.1.1, а расшифровка полей, представленных на изображении, приведена в таблице 1.2.1.1.

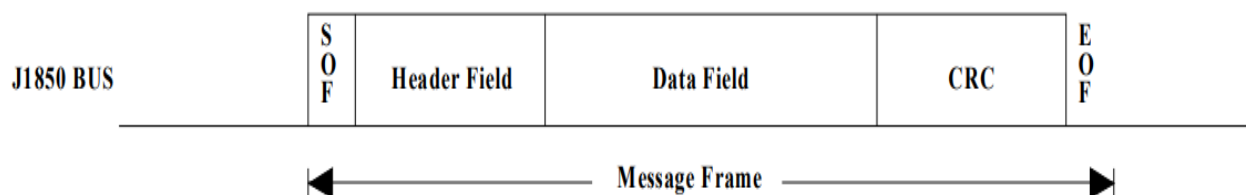


Рисунок 1.2.1.1 – Формат кадра SAE J1850



Таблица 1.2.1.1 – Расшифровка полей кадра SAE J1850

Название поля	Длина, бит	Назначение
SOF	-	Обозначение начала кадра. Устанавливается высокий уровень сигнала на шине на период 200мс
Header Field	8-24	Поле заголовка. Содержит критичную информацию о кадре: Длину поля заголовка, количество байт данных, приоритет, тип адресации и пр.
Data Field	64-88	Данные
CRC	8	Циклический контроль избыточности
EOF	-	Обозначение начала кадра. Устанавливается низкий уровень сигнала на шине на период 200мс

### 1.2.2. Коммуникационный протокол KWP-2000

Keyword Protocol 2000, также известный как KWP2000, является коммуникационным протоколом, используемым для систем бортовой диагностики транспортных средств. Данный протокол охватывает прикладной уровень в модели OSI компьютерной сети. Протокол стандартизирован Международной организацией по стандартизации как ISO-14230. KWP2000 также охватывает сеансовый уровень в модели OSI с точки зрения начала, поддержания и завершения сеанса связи.

Данные передаются по одной двунаправленной последовательной линии связи, называемой K-Line. Кроме того, на некоторых устройствах, существует дополнительная L-линия для пробуждения. Скорость передачи данных

составляет от 1,2 до 10,4 килобод, передаваемое сообщение может содержать до 255 байт в поле данных[11-14].

Для начала передачи данных, необходимо провести инициализацию линии. Эту процедуру можно произвести двумя методами: методом быстрой инициализации, либо методом 5-бод. Каждый из них критичны к временным интервалам и требуют обязательного их соблюдения.

Протокол KWP2000 также совместим со стандартом ISO-11898 (Controller Area Network), поддерживающим более высокие скорости передачи данных до 1 Мбит/с. CAN становится все более популярной альтернативой K-line, поскольку шина CAN обычно присутствует в современных автомобилях и, таким образом, устраняет необходимость установки дополнительного физического кабеля.

Наиболее распространенным является использование KWP2000 на CAN с транспортными/сетевыми слоями стандарта ISO-15765. Кроме того, использование KWP2000 на шине CAN не требует специальной функции пробуждения.

Формат кадра, передаваемого по протоколу KWP2000, приведен на рисунке 1.2.2.1, а расшифровка полей кадра, представленных на изображении, приведена в таблице 1.2.2.1.

Header				Data bytes					Checksum
Fmt	Tgt	Src	Len	Sld	...	Data	...	CS	

Рисунок 1.2.2.1 – Формат кадра KWP2000

Таблица 1.2.2.1- Расшифровка полей кадра KWP2000

Название поля	Длина, байт	Назначение
Format	1	Байт, определяющий формат сообщения
Target	1	Байт, определяющий адрес приемника сообщения

### Продолжение таблицы 1.2.2.1

Source	1	Байт, определяющий адрес источника сообщения
Length	1	Байт, определяющий длину сообщения
SId	1	Байт, определяющий тип передаваемых данных. Является частью байтов данных
Data	1-254	Данные
Checksum	1	Байт контрольной суммы

### 1.2.3. Коммуникационный протокол CAN2.0A

CAN – это промышленная сеть, которая оказалась удобной для использования в электронных системах автомобиля. На всех автомобилях после 2008 года выпуска по умолчанию должен присутствовать этот тип связи.

CAN (Controller network) – Сеть, состоящая, как правило, из двух проводов (витой пары), обеспечивающая подключение к ней различных устройств ввода-вывода, датчиков, исполнительных механизмов, предназначенная для обмена данными между устройствами без применения компьютера-хоста. Для того чтобы устройства обменивались сообщениями между собой, были разработаны несколько протоколов, например CAN 2.0A, CAN 2.0B, CAN FD 1.0 и пр. Для подключения к сети CAN необходимо использовать специализированное устройство - CAN трансивер. CAN-трансивер необходим для защиты подключенных устройств от внезапных неполадок в бортовой сети автомобиля и для согласования логических уровней сигналов. Скорость передачи данных по данному интерфейсу может составлять до 1Мбит/с и выбирается исходя из длины сети. Максимальное число

подключенных устройств к шине определяется нагрузочной способностью трансиверов [8].

Шина CAN является асинхронной с типом доступа, допускающим разрешение коллизии. Передача данных происходит с помощью кадров. Для абстрагирования от среды передачи данных, в спецификации CAN приняты термины «рецессивный» бит (логическая единица) и «доминантный» бит (логический ноль). Принято, что при одновременной передаче рецессивного бита и доминантного бита, будет принят доминантный.

Существуют четыре типа кадров передаваемых по шине CAN [9]:

- data frame (кадр данных) – предназначен только для передачи данных;
- remote frame (кадр удаленного запроса) – предназначен для отправки запроса какому-либо устройству;
- overload frame (кадр перегрузки) – кадр, посылаемый устройством, которое не может в данный момент принять данные;
- error frame (кадр ошибки) – кадр, посылаемый устройствами, при обнаружении ошибки в сети.

Структура кадра, передаваемого по шине CAN согласно спецификации CAN2.0A представлена в таблице 1.2.1.1 [10]:

Таблица 1.2.1.1 – Структура кадра CAN2.0A

Название поля	Длина (бит)	Назначение
Начало кадра	1	Предназначено для извещения всех устройств, находящихся в сети о начале передачи данных передающим устройством

Продолжение таблицы 1.2.1.1

Название поля	Длина (бит)	Назначение
Идентификатор	11	Уникальный идентификатор передающего устройства. Представляет собой приоритет сообщения во время арбитража.
Запрос на передачу	1	Должен быть доминантным для кадров данных и рецессивным для кадров удаленного запроса.
Бит расширенного идентификатора	1	Должен быть доминантным.
Зарезервированный бит	1	Должен быть доминантным, но принимается и доминантным и рецессивным.
Код длины данных	4	Код, указывающий количество байт данных в сообщении.
Поле данных	0-64	Содержит данные.
Циклический контроль избыточности	15	Алгоритм нахождения контрольной суммы сообщения.
CRC delimiter	1	Разделитель. Должен быть рецессивным.
Слот подтверждения приема	1	Передатчик отправляет рецессивный бит(1), в случае успешного приема, приемник устанавливает доминантный (0) бит.
АСК delimiter	1	Разделитель. Должен быть рецессивным.
Конец кадра	7	Поле конца сообщения.

## 1.2.4. Обзор микроконтроллеров

Микроконтроллер - главная составляющая часть будущего программно-аппаратного комплекса. Он должен определять будущий диагностический протокол, отправлять запросы электронным блокам автомобиля и принимать ответы на них, а также отображать на дисплее полученную информацию.

Сегодня на рынке присутствуют множество семейств микроконтроллеров, и выбрать подходящее семейство не составляет труда. Так как от конечного устройства требуется, чтобы оно могло работать с CAN-шиной, то проще всего использовать микроконтроллер с заведомо присутствующим CAN-интерфейсом. Это решение является выигрышным, так как не будут тратиться дополнительные ресурсы центрально процессора (у большинства микроконтроллеров CAN-интерфейс обладает собственным обработчиком кадров).

В качестве претендентов можно выделить 3 семейства микроконтроллеров:

- STM32 производства STMicroelectronics;
- PIC производства Microchip Technology;
- AVR производства Microchip Technology.

AVR - семейство микроконтроллеров, разработанное в 1996 году корпорацией Atmel. Все микроконтроллеры из этого семейства построены по гарвардской архитектуре, согласно которой данные и программа хранятся отдельно друг от друга. Микроконтроллеры имеют по 32 регистра общего назначения, до 86 интерфейсов ввода/вывода общего назначения, до 3 источников внешних прерываний, внутреннюю память программ до 256Кб [17].

Наиболее широкое распространение получили три подсемейства микроконтроллеров AVR:

- tinyAVR: обладают наименьшим объемом памяти, небольшим количеством интерфейсов ввода-вывода, малым числом периферийных устройств. Идеально подходят для решения мелких задач;
- megaAVR: обладают большим объемом памяти, значительно бОльшим количеством интерфейсов ввода-вывода по сравнению с подсемейством tinyAVR, более развитой периферией. Подходят для решения сложных задач;
- xmegaAVR: самое мощное подсемейство. Обладает еще бОльшим количеством периферии, памяти программ, интерфейсов ввода-вывода, чем подсемейство megaAVR. Используется для решения промышленных задач, требующих высокой скорости и точности вычислений.

РISC-семейство микроконтроллеров, производящихся по гарвардской архитектуре компанией Microchip Technology. В линейке присутствуют как 8, так 16 и 32-битные микроконтроллеры.

Микроконтроллеры RISC пользуются спросом как у промышленных разработчиков, так и у любителей из-за их доступности, низкой стоимости, большой базе пользователей, обширной коллекции заметок о приложениях, наличию недорогих или бесплатных инструментов разработки, программированию flash-памяти по последовательному интерфейсу[18].

STM32 - семейство 32-разрядных микроконтроллеров, выполненных по RISC архитектуре, на базе процессора Arm Cortex - M. Это семейство разрабатывалось с целью предоставления будущим пользователям новых возможностей, редко использовавшихся ранее. Данная линейка предлагает продукты, сочетающие в себе невысокую стоимость, цифровую обработку сигналов, работу при небольшом напряжении питания, высокую

производительность, сохраняя при этом простоту разработки и легкость интеграции [19].

Для разработки и поддержки проектов для микроконтроллеров предоставляется широкий ассортимент платного и бесплатного программного обеспечения.

Вышеперечисленные факты делают данное семейство продуктов идеальным как для небольших проектов, так и для больших платформ.

В рамках данной выпускной работы будет применяться микроконтроллер семейства STM32 ввиду его высокой производительности, возможности подключения большого количества периферийных устройств по различным протоколам, в том числе и CAN 2.0A, постоянного развития этих микроконтроллеров, а также невысокой стоимости.

### **1.2.5. Обзор инструментария**

В настоящий момент существует огромное количество сред разработки, используемых для написания программ для микроконтроллеров STM32. Наиболее популярными из них являются: Keil Uvision, IAR Embedded Workbench, Coocox IDE, STM32 Cube IDE. Каждая из них является по-своему уникальной [20].

Keil Uvision – это оконная платформа для разработки программного обеспечения. Она сочетает в себе современный и надежный редактор исходного кода, менеджер проекта и инструмент создания объектов. Данная платформа объединяет в себе все инструменты, необходимые для разработки встраиваемых приложений, включая компилятор C/C++, макро ассемблер, компоновщик, генератор шестнадцатеричных файлов.  $\mu$ Vision помогает ускорить процесс разработки встраиваемых приложений, предоставляя следующие возможности:



- полнофункциональный редактор исходного кода;
- база данных устройств для настройки инструмента разработки;
- менеджер проектов для создания и сопровождения ваших проектов;
- утилита Flash programming для загрузки прикладной программы в Flash ROM устройства;
- ссылки на руководства, справочные материалы, документацию на устройства и руководства пользователя;
- встроенная функциональная утилита Make Utility для сборки, компиляции и связывания встраиваемых приложений;
- диалоговые окна для всех параметров среды разработки;
- настоящий интегрированный отладчик исходного уровня и уровня ассемблера с высокоскоростным процессором и периферийным симулятором;
- современный интерфейс GDI для отладки программного обеспечения на целевом устройстве.

IDE и отладчик  $\mu$ Vision являются центральной частью цепочки инструментов разработки Keil и имеют множество функций, которые помогают программисту быстро и успешно разрабатывать встраиваемые приложения. Инструменты Keil просты в использовании и гарантированно помогут своевременно достичь поставленных целей.

$\mu$ Vision предлагает режим сборки для создания приложений и режим отладки для отладки приложений. Приложения могут быть отлажены с помощью встроенного симулятора  $\mu$ Vision или непосредственно на аппаратном обеспечении, например Ulink Debug. Разработчики также могут использовать другие адаптеры или внешние сторонние инструменты для анализа приложений [21].

У данной среды разработки очень много достоинств, но есть существенный минус: бесплатная версия хоть и обладает полным функционалом, но максимальный объем программы, которую можно загрузить в устройство, равен 32Кб.

В данной работе используется среда разработки Keil Uvision, так как она имеет большее по сравнению с другими количество функций, а также ранее была приобретена платная версия данной среды разработки.

### **1.2.6. Устройства согласования логических уровней**

Для решения поставленных задач необходимо отправлять запросы по диагностической шине автомобиля и получать ответы на них, извлекать из ответов необходимую информацию и обрабатывать ее. Для этого необходимо решить проблему согласования логических уровней. Дело в том, что в бортовой сети автомобиля напряжение может находиться в диапазоне от 7 до 30 Вольт (последнее применимо к грузовым автомобилям), а напряжение питания микроконтроллера, и, соответственно, его пинов ввода-вывода как правило не превышает 5 Вольт. Решением этой проблемы являются микросхемы согласования логических уровней. Таких микросхем понадобится как минимум три: для каждого интерфейса по одной.

В качестве устройства согласования уровней на шине CAN применяются CAN-трансиверы. CAN-трансивер- это компонент, обеспечивающий связь между цифровыми устройствами сети. CAN-трансиверы часто используются, когда необходимо обеспечить связь в условиях жесткой физической среды. Например, при высоких температурах, высокой влажности, наличии помех. Трансивер может передавать и принимать данные в другие части схемы и от них. Многие электронные устройства имеют трансивер в той или иной форме.

Из-за большого разнообразия потребностей, которые могут быть удовлетворены с помощью трансиверов, в настоящее время на рынке существует множество разновидностей. Они различаются типом сигнала, который они посылают (аналоговый или цифровой), независимо от того, предназначены ли они для портативного использования, количества требуемой мощности и других особенностей.

Основные характеристики трансивера CAN заключаются в том, что он обладает высокой мощностью, портативен, и может быть как цифровым, так и аналоговым. Он может работать в пределах большого диапазона температур. Количество энергии, необходимой для его работы, варьируется примерно от 3 до 5 вольт. Стандартный приемопередатчик CAN обычно сконструирован так, чтобы нормально функционировать даже при скачках напряжения и потере заземления. Большинство других электронных устройств не могут работать адекватно в этих условиях.

Несмотря на разнообразие продуктов на рынке, для достижения поставленной цели потребуется подобрать трансивер с параметрами, подходящими для связи с большинством транспортных средств.

Для передачи информации по протоколу KWP-2000 требуется соответствующая микросхема. Микроконтроллер будет отправлять информацию по интерфейсу UART, а микросхема будет преобразовывать эту информацию в понятный вид для электронного блока управления. На рынке присутствует большое количество микросхем передачи данных по протоколу KWP2000. Наиболее распространенными из них являются L9637D производства STMicroelectronics, MC33199D производства Motorola inc. и пр.

Для передачи информации по протоколу SAE J1850 можно применить либо встроенную возможность широтно-импульсной модуляции в микроконтроллере, либо специальную микросхему. Для защиты

проектируемого устройства от непредвиденных ситуаций в бортовой сети автомобиля нужно выбрать второй вариант.

### **1.2.7. Стабилизатор напряжения**

Для обеспечения безотказной работы программно-аппаратного комплекса, необходимо, чтобы питание микроконтроллера и периферии было качественным. Для этого применяются стабилизаторы напряжения.

Насколько известно, микроконтроллеры очень требовательны к питанию, и малейший скачок напряжения может вывести микроконтроллер из строя. Таким образом, к стабилизатору должны быть предъявлены особые требования. Наибольшую популярность в последнее время приобрели линейные стабилизаторы напряжения. Линейный стабилизатор напряжения представляет из себя делитель напряжения, на вход которого подаётся нестабильное напряжение, а с нижнего плеча делителя снимается стабилизированное (выходное) напряжение. Стабилизация осуществляется путём изменения сопротивления одного из плеч делителя: сопротивление постоянно поддерживается таким, чтобы напряжение на выходе стабилизатора находилось в установленных пределах.

При большом отношении величин входного/выходного напряжений линейный стабилизатор обладает низким коэффициентом полезного действия, так как большая часть входной мощности рассеивается на регулирующем элементе в виде тепла, поэтому регулирующий элемент в стабилизаторах такого типа и повышенной мощности должен рассеивать значительную мощность, то есть должен быть установлен на радиатор нужной площади.

Преимуществами линейных стабилизаторов являются небольшое количество используемых электронных компонентов простота, отсутствие помех [22].

### **1.2.8. Язык программирования микроконтроллера**

Существует три наиболее популярных языка программирования микроконтроллеров семейства STM32: C, C++, Assembler.

Assembler является языком низкого уровня. Программа на нем пишется с помощью машинных команд. Данный язык позволяет написать программу, которая будет работать быстро и занимать минимальное место в памяти микроконтроллера. В некоторых случаях альтернативы языку Assembler не существует, однако и он имеет несколько недостатков. Несмотря на небольшой объем памяти устройства, занимаемый программой, код зачастую получается громоздким и труднопонимаемым, что в дальнейшем может негативно сказаться на дальнейшей его поддержке. Помимо этого, для создания программы на языке Assembler, необходимо хорошо знать архитектуру устройства, для которого пишется программа, или же постоянно прибегать к помощи технической документации.

На основе вышесказанного напрашивается вывод о том, что Assembler подходит для разработчиков с большим стажем работы с электроникой, а также в случае ограниченности ресурсов устройства.

Языки программирования C/C++ относятся к языкам высокого уровня. Программа на этих языках более понятна для человека. Помимо этого, данные языки обладают великим множеством разнообразных библиотек и программных сред, позволяющих создавать более компактный код.

Фактически, языки C/C++ стали основными для разработки управляющих программ для микроконтроллеров. При написании кода также требуется

учитывать архитектуру микроконтроллера. Однако, за все плюсы приходится чем-то платить. В данном случае - памятью микроконтроллера. Программа, написанная на языках C/C++ занимает, как правило, больше места, чем абсолютно такая же программа, написанная на языке Assembler. Причина кроется в несовершенстве компиляторов.

Невозможно не упомянуть такие языки программирования как Pascal и Basic. Они хоть и легки в изучении, но в настоящий момент становятся все менее востребованными.

В рамках выпускной квалификационной работы все программы будут написаны исключительно на языке C.

### **1.2.9. Библиотека, используемая для написания программы**

При программировании микроконтроллеров семейства STM32 разработчики чаще всего прибегают к помощи таких библиотек, как CMSIS, HAL, SPL, LL и пр. Рассмотрим более подробно возможности каждой из них.

CMSIS (The Cortex Microcontroller Software Interface Standard) - это библиотека, разработанная для микроконтроллеров, основанных на процессорах Arm Cortex. Ее программные интерфейсы сокращают время обучения для разработчиков микроконтроллеров и ускоряют время выхода на рынок новых устройств. CMSIS предоставляет интерфейсы для процессоров и периферийных устройств, операционных систем реального времени и компонентов связующего ПО (англ. middleware). Ее использование является профессиональным способом написания программ, потому что предполагает прямую работу с регистрами микроконтроллера. Соответственно, для работы с данной библиотекой требуется постоянное чтение технической документации прилагаемой к микроконтроллеру [23].

SPL (Standard Peripherals Library) - библиотека, разработанная компанией ST Microelectronics на языке C для своих микроконтроллеров. Она включает в себя структуры, макросы и функции для облегчения работы с периферией микроконтроллера. Библиотека документирована, включает примеры по каждому периферийному устройству, полностью поддерживает CMSIS [24].

Библиотека HAL (Hardware Abstraction Layer) состоит из драйверов верхнего уровня. Драйверы HAL представляют из себя комплект универсальных, многофункциональных, и одновременно простых интерфейсов API, предназначенных для взаимодействия микроконтроллера с верхним слоем программного обеспечения. Библиотека HAL была создана для применения такой архитектуры программирования, когда необходимые функции выполняются верхним слоем приложения, за счет применения промежуточного уровня HAL. При использовании такой архитектуры программирования, верхний уровень приложения не «привязан» к микроконтроллеру, так как обращается к ресурсам микроконтроллера только через библиотеку драйверов HAL. Такая структура пользовательского приложения улучшает поддержку кода, и гарантирует его легкую переносимость на другие устройства семейства STM32. Драйверы HAL предоставляют пользователю полный набор готовых к использованию API, которые упрощают реализацию конечного приложения.

Драйверы слоя HAL имеют возможность обнаружения ошибок во время выполнения (failure detection during runtime). HAL проверяет значения всех переменных, подающихся на вход функций. Такая динамическая проверка способствует повышению безотказности встроенного программного обеспечения. Обнаружение ошибок во время выполнения программы также содействует ускорению процессов разработки приложений и отладки [25].

Библиотека драйверов LL (low-layer) предназначена для обеспечения быстрого, экспертно-ориентированного уровня, который ближе к аппаратному обеспечению, чем библиотека HAL. В отличие от HAL, LL API не

предусмотрены для использования в периферийных устройствах, где оптимизированный доступ не является ключевой функцией. С помощью драйверов библиотеки LL можно работать напрямую с регистрами и периферией микроконтроллера, однако, для ее использования, все равно необходимо иногда обращаться к технической документации на устройство, для которого разрабатывается программа [26].

В рамках данной выпускной квалификационной работы все программы будут написаны с применением библиотеки CMSIS, т.к. написанный с ее применением код хоть и не всегда компактен, но работает быстрее, чем написанный на прочих библиотеках, а также занимает меньше памяти микроконтроллера.

### **1.3. ВЫВОД**

Были рассмотрены существующие аналоги программно-аппаратного комплекса, произведен обзор диагностических протоколов обмена информацией, произведен обзор семейств микроконтроллеров. Также был обоснован выбор каждого программного и аппаратного средства, применяющегося при разработке программно-аппаратного комплекса.

Обзор аналогов выявил слабые стороны, имеющиеся у всех исследованных диагностических сканеров, соответственно, требуется разработать программно-аппаратный комплекс, лишенный подобных недостатков.

Микроконтроллером в программно-аппаратном комплексе будет являться STM32, применяемый язык программирования - C, применяемая среда разработки - Keil Uvision. Также необходимо будет использовать микросхемы для связи с интерфейсами CAN, KWP2000, SAE J1850.



## **2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ**

Для реализации программно-аппаратного комплекса понадобится следующий список компонентов:

- микроконтроллер, который будет отправлять запросы электронным блокам, получать ответы на них, обрабатывать эту информацию и выводить результат на LCD-дисплей;
- CAN-трансивер для связи с контроллерами с применением интерфейса CAN2.0A;
- микросхема интерфейса KWP2000 для связи с контроллерами с применением интерфейса KWP2000;
- микросхема интерфейса SAE J1850 для связи с контроллерами с применением интерфейса SAE J1850;
- LCD – дисплей для отображения параметров, полученных от электронных блоков автомобиля;
- тактовые кнопки для навигации по меню программы микроконтроллера;
- штекер OBD-II для подключения к разъему диагностики;
- соединительные провода.

### **2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ**

На рисунке 2.1 представлена схема взаимодействия объектов системы. Она состоит из двух модулей и микроконтроллера.

Модуль взаимодействия с пользователем отвечает за выбор режима работы комплекса: отображение параметров в режиме реального времени, либо чтение неисправностей.

Модуль обмена информацией, в свою очередь, отвечает за коммуникацию микроконтроллера с электронными блоками транспортного средства посредством микросхем соответствующих интерфейсов.

После выбора режима работы, микроконтроллер производит отправку, прием, обработку информации и вывод ее на LCD-дисплей.

Диагностический разъем является одновременно и связующим звеном и источником питания, так как в нем установлен линейный стабилизатор напряжения.

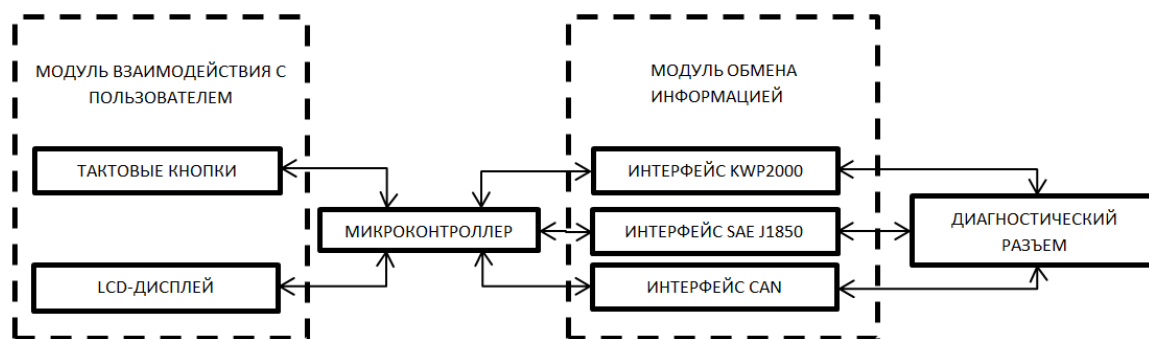


Рисунок 2.1 Схема взаимодействия объектов системы

На основе рисунка 2.1 определим функциональные требования будущего программно-аппаратного комплекса:

- чтение диагностических кодов неисправностей;
- вывод на экран полученных кодов неисправностей;
- расшифровка кодов неисправностей;
- удаление кодов неисправностей из памяти электронного блока управления;

- управление исполнительными механизмами ЭСУД.

## **2.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ**

Программно-аппаратный комплекс, исходя из своего назначения, будет эксплуатироваться в широком диапазоне сервисного обеспечения, как в условиях высококачественного сервиса у официальных дилеров, так и в условиях индивидуального и частного пользования, поэтому при разработке комплекса была учтена возможность работы в условиях повышенной вибрации, повышенной влажности, влияния различных электромагнитных помех от системы зажигания.

### **2.2.1. Требования, предъявляемые к устройству:**

- возможность подключения к диагностическому разъему автомобиля;
- размеры не более 100мм\*150мм\*50мм;
- масса не более 500гр;
- защита от воды, масла, бензина и их производных, пыли, грязи, ударов;
- напряжение питания 12-24В;
- защита от короткого замыкания и помех со стороны системы зажигания.

### **2.2.2. Требования, предъявляемые к диагностируемому автомобилю:**

- наличие разъема диагностики OBD-II;
- исправность диагностических интерфейсов (целостность проводов, электронных блоков);
- год выпуска от 1998.

### **2.2.3. Требования, предъявляемые к диагносту:**

- знание принципов работы и устройства двигателя внутреннего сгорания;
- знания в области электроники и схемотехники;
- базовые знания в области автомобильной диагностики.

### 3. ПРОЕКТИРОВАНИЕ

#### 3.1. АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ

Главным компонентом программно-аппаратного комплекса будет являться микроконтроллер STM32F103C8T6 в корпусе LQFP48. При таком исполнении микроконтроллер имеет 48 контактов, 4 из которых предназначены для питания, еще 4 являются нулевыми шинами питания Vss, два контакта предназначены для подключения внешнего кварцевого резонатора, еще два предназначены для выбора режима работы при загрузке микроконтроллера. Таким образом, можно использовать только 32 контакта для работы с микроконтроллером. Назначение выводов микроконтроллера STM32F103C8T6 в корпусе LQFP48 приведено на рисунке 3.1.1.

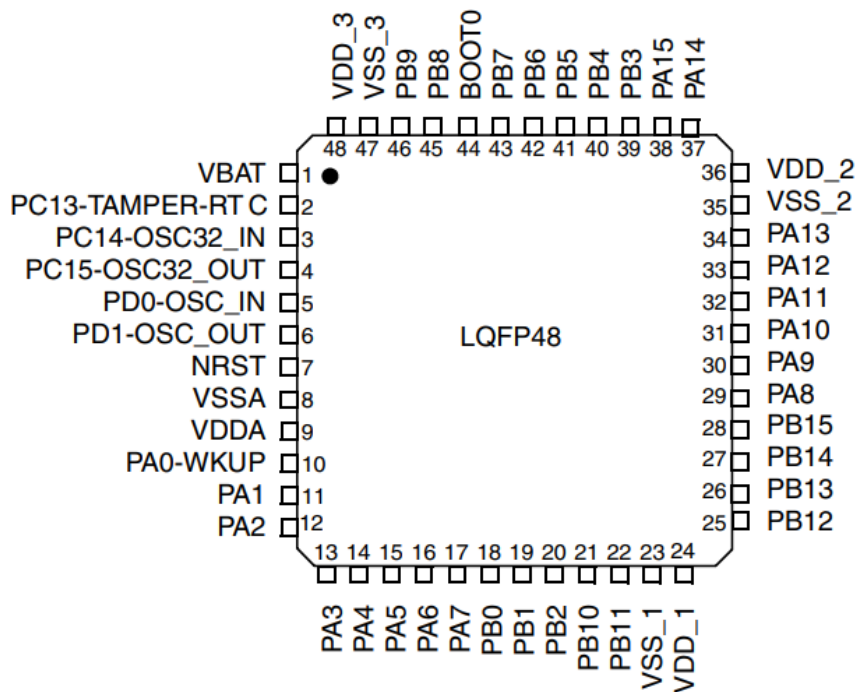


Рисунок 3.1.1 – Назначение выводов микроконтроллера STM32F103C8T6

Составим схему подключения модулей к микроконтроллеру. LED-дисплей подключим по шине I2C, для микросхемы интерфейса KWP2000

применим интерфейс UART, для подключения CAN-трансивера применим заведомо присутствующий модуль bxCAN, для подключения микросхемы интерфейса SAE J1850 используем интерфейс UART, присоединим 4 тактовые кнопки для навигации в меню программы. Также для обеспечения схемы питанием, применим стабилизатор напряжения. Схема устройства приведена на рисунке А.1 в приложении А.

Порты микроконтроллера, и соответствующие им назначенные функции приведены в таблице 3.1.1.

Таблица 3.1.1 - Порты микроконтроллера и соответствующие им функции

Название порта микроконтроллера	Назначение
PA11	CAN RX
PA12	CAN TX
PB12	CS LCD
PB13	E LCD
PB15	R/W LCD
PB10	SAE J1850 TX
PB11	SAE J1850 RX
PA0	Тактовая кнопка 1
PA1	Тактовая кнопка 2
PA2	Тактовая кнопка 3
PA3	Тактовая кнопка 4
PA9	K-line TX
PA10	K-line RX

## 3.2. АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧ

Для подключения диагностического тестера к электронным блокам, необходимо определить, какие линии связи присутствуют. Это можно сделать разными способами, но самым популярным из них является метод перебора. Согласно стандарту сообщества автомобильных инженеров SAE J1978 [27], любой диагностический тестер должен поддерживать коммуникацию с 5 линиями связи, а также автоматически определять имеющиеся линии связи в бортовой сети автомобиля. В рамках текущей выпускной квалификационной работы был разработан алгоритм автоматического определения присутствующих линий связи путем перебора: тестер отправляет запрос инициализации и ожидает на него ответ. Если ответа нет, предполагается, что линия связи была разорвана, либо отсутствует в сети автомобиля. Тестер автоматически переходит к проверке последующих линий связи. Если ни одна линия связи не была обнаружена в автомобиле (подобного рода ситуация встречается на практике довольно часто), пользователю предлагается выбрать линию связи самостоятельно. В случае отсутствия всех линий связи на экран выводится соответствующее сообщение.

Схема алгоритма автоматического определения присутствующих линий связи представлена на рисунке 3.2.1.

После определения присутствующих линий связи необходимо отправлять запросы по ним к электронным блокам и получать ответы на них. Согласно функциональным требованиям, в дальнейшем должны быть определены 4 режима работы тестера:

- чтение диагностических кодов неисправностей и отображение их на экране;

- удаление кодов неисправностей из памяти электронного блока управления;
- управление исполнительными механизмами ЭСУД;
- отображение параметров ЭСУД в режиме реального времени.

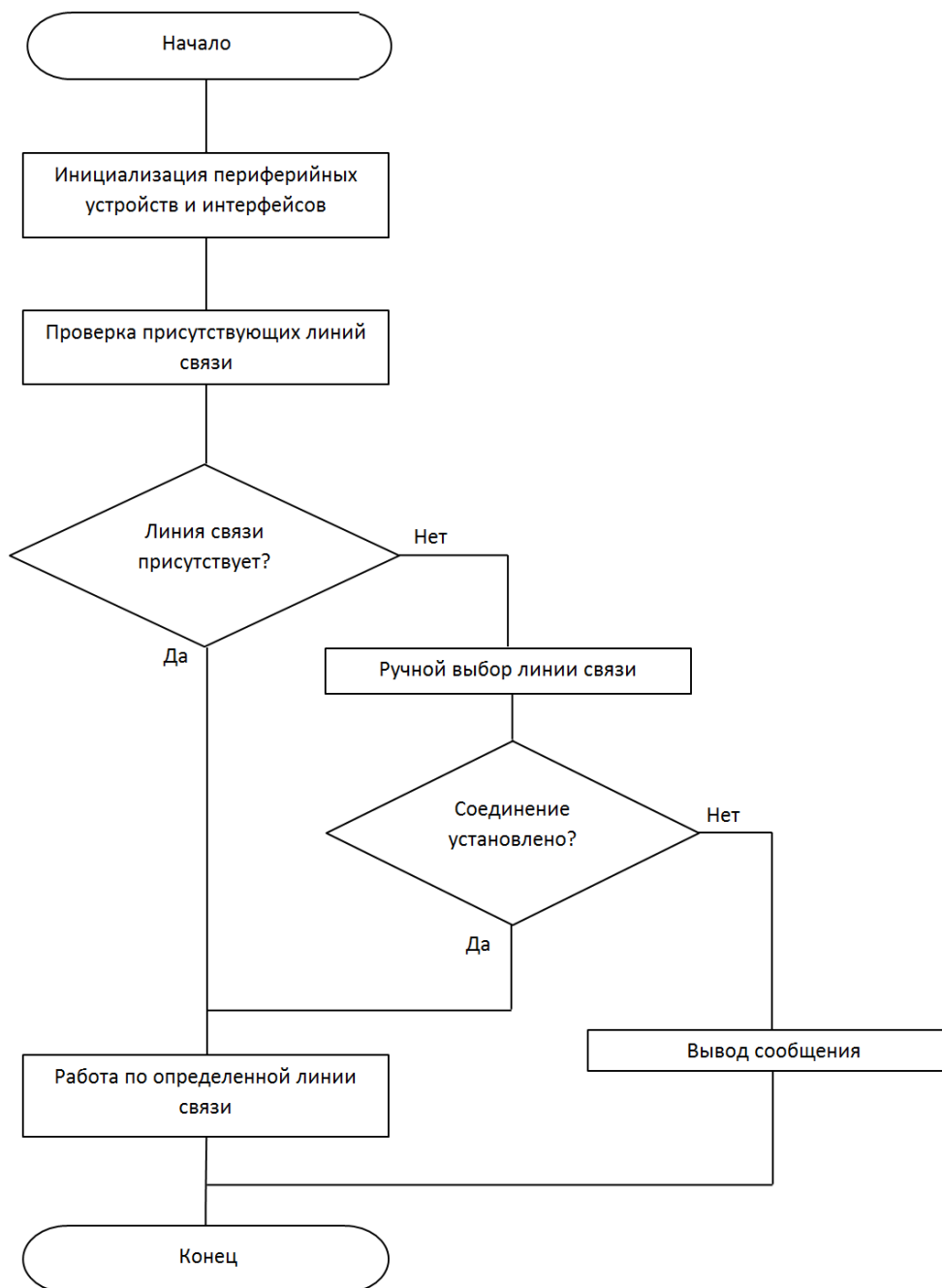


Рисунок 3.2.1 - Алгоритм автоопределения линий связи.



Алгоритмы оставшихся режимов работы: чтения ошибок, управления исполнительными механизмами, отображения списка параметров ЭСУД в режиме реального времени представлены соответственно на рисунках 3.2.2-3.2.4.

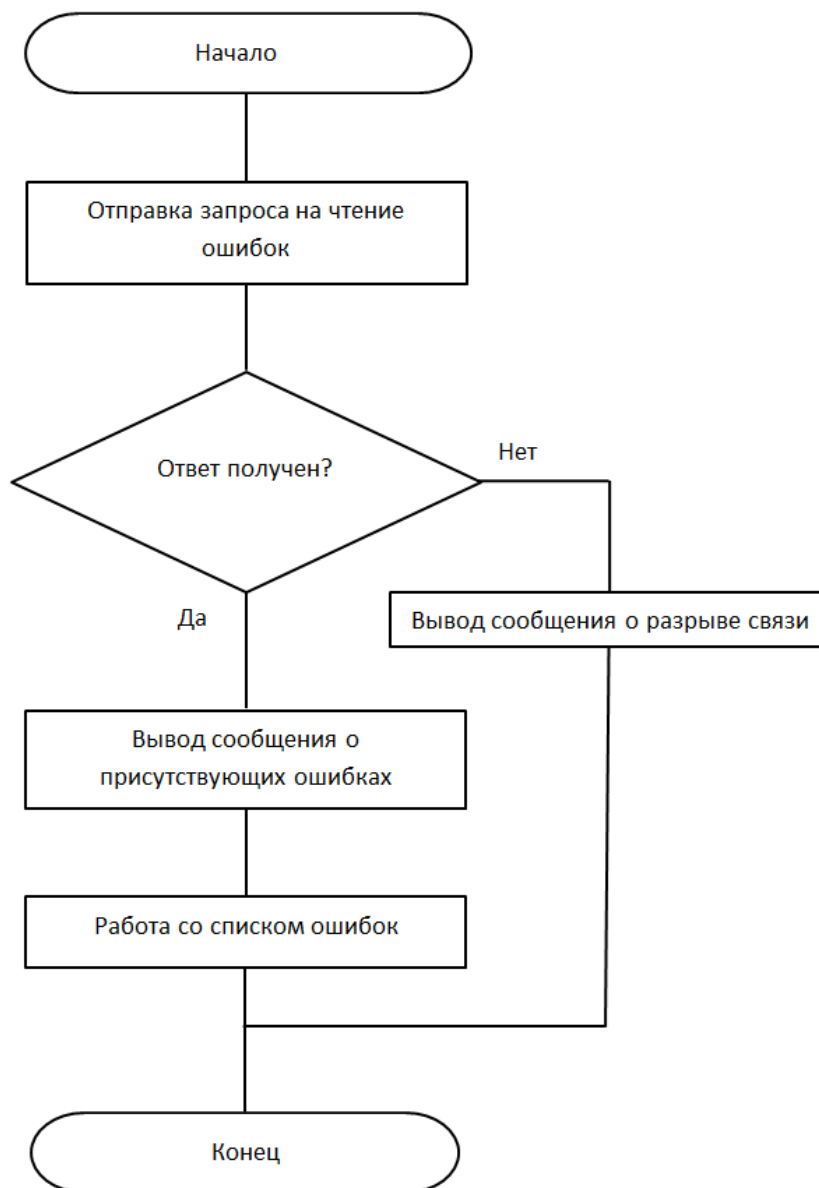


Рисунок 3.2.2 - Алгоритм чтения ошибок

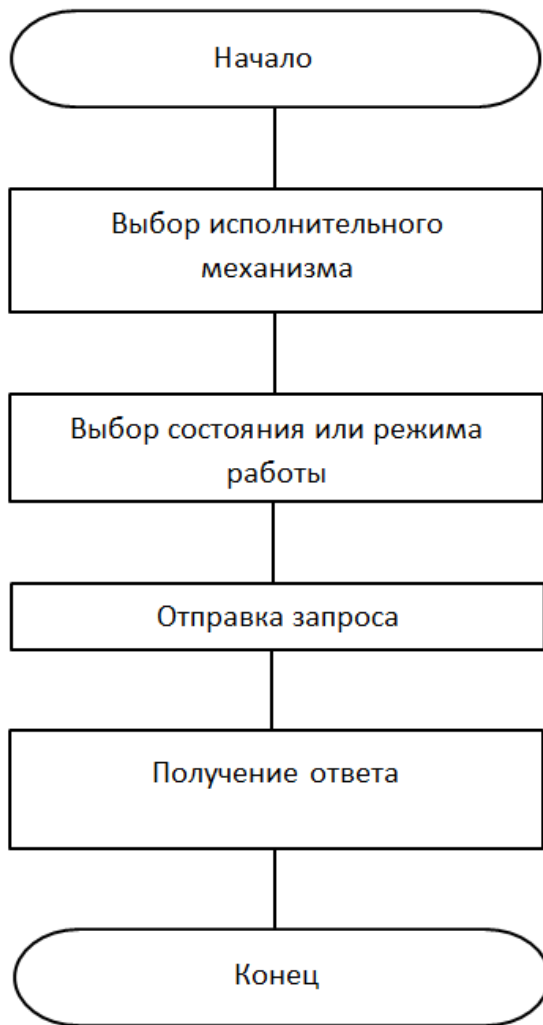


Рисунок 3.2.3 - Алгоритм управления исполнительными механизмами

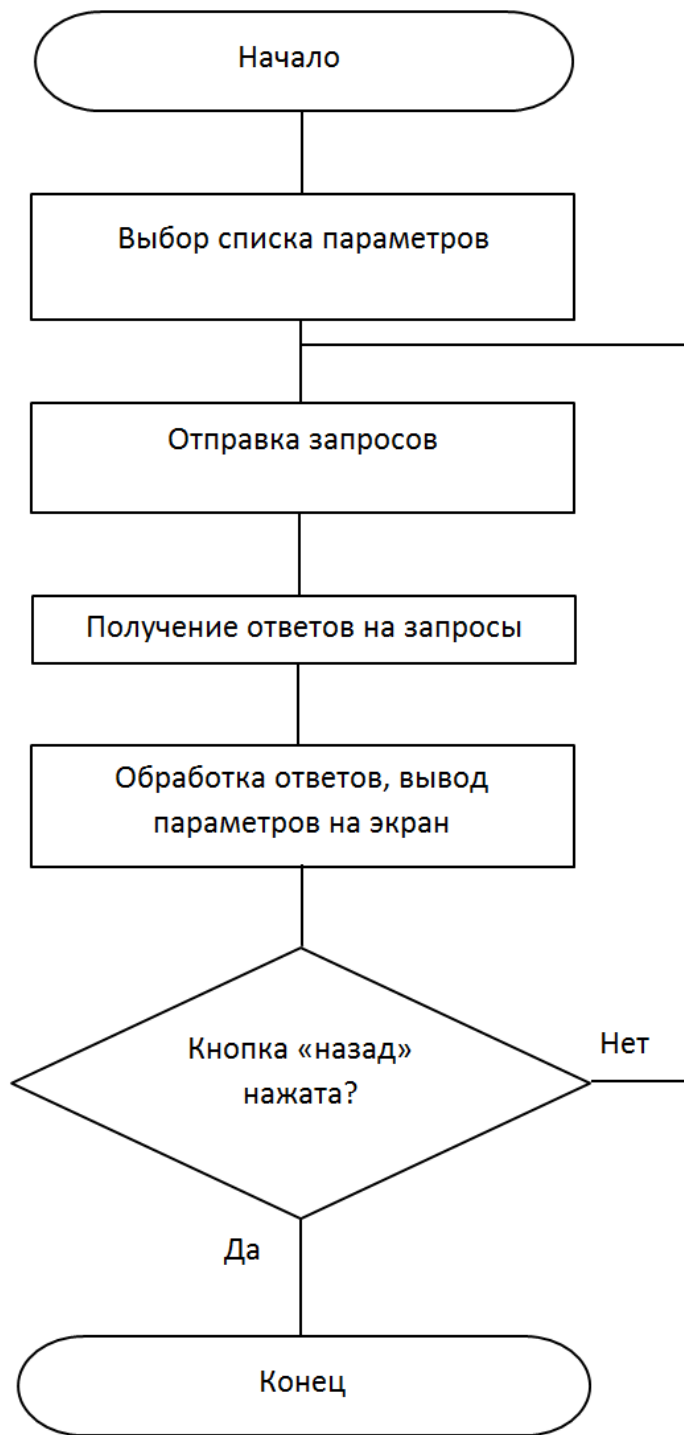


Рисунок 3.2.4 - Алгоритм отображения параметров ЭСУД в режиме реального времени

### 3.3. ОПИСАНИЕ ДАННЫХ

Для того чтобы получить диагностическую информацию, требуется отправить запрос электронному блоку автомобиля. Для унификации запросов, был разработан стандарт SAE J1979, который описывает режимы работы тестера и коды, используемые для получения диагностической информации [28]. Всего доступно 10 режимов работы, они приведены в таблице 3.3.1.

Таблица 3.3.1 – Режимы работы тестера

Номер режима	Номер режима (Hex)	Описание
1	0x01	Чтение текущих параметров систем управления
2	0x02	Чтение параметров систем управления в момент возникновения неисправности
3	0x03	Считывание сохраненных кодов неисправностей
4	0x04	Очистка кодов неисправностей и сохраненных значений
5	0x05	Считывание и просмотр результатов тестирования датчиков кислорода (не для CAN шины)
6	0x06	Считывание результатов тестов датчиков, отвечающих за экологические нормы (считывание и просмотр результатов теста датчиков кислорода только для CAN шины)
7	0x07	Запрос ошибок, присутствующих в системах управления в текущий момент
8	0x08	Управление исполнительными механизмами
9	0x09	Запрос информации о транспортном средстве
10	0x0A	Запрос списка ошибок, которые были удалены ранее

Выбрав режим работы нужно получить диагностическую информацию, либо воздействовать на исполнительные механизмы. Это возможно благодаря кодам запроса диагностической информации. Они называются PID (Parameter ID). Данный код вместе с режимом работы позволяет получить требуемую информацию.

Чтобы расшифровать, какие PID поддерживает электронный блок управления, нужно отправить ему запрос с соответствующим режимом работы и PID = 0x00. В ответ электронный блок отправит список поддерживаемых параметров. Например, выбрав режим работы 0x01 и Parameter ID 0x00, был получен ответ 0x318AF1EB. После перевода этого числа в двоичную систему счисления единицам будут соответствовать поддерживаемые параметры, нулям - не поддерживаемые параметры. Расшифровка ответа в графическом виде показана на рисунке Б.1 в приложении Б. Описание нестандартного списка кодов запроса диагностической информации можно получить, обратившись к стандарту SAE J2190 [29].

Также следует отметить, что при использовании протоколов KWP2000 и SAE J1850 физический адрес тестера согласно стандарту SAE J2178 равен 0xF1, а физический адрес блока управления двигателем равен 0x10. При использовании протокола CAN 2.0A ID кадра тестера равен 0x7DF, а ID кадра ответа главного электронного блока равен 0x7E8 [30].

## **4. РЕАЛИЗАЦИЯ**

### **4.1. РЕАЛИЗАЦИЯ ТЕСТОВОГО УСТРОЙСТВА**

Чтобы проверить работоспособность системы, необходимо собрать устройство с применением вышеупомянутых плат и микросхем. Покупка компонентов производилась в магазинах «ЧИП и ДИП», «Проконтакт 74», «Промэлектроника» [31-33]. Не обнаруженные на территории Российской Федерации компоненты были заказаны с сайта «Aliexpress» [34].

#### **Микроконтроллер STM32F103C8T6**

Основой тестового устройства выступает плата с микроконтроллером STM32F103C8T6, известная как «Blue Pill». Большим плюсом следует считать тот факт, что на ней с завода присутствуют такие необходимые компоненты как стабилизатор напряжения, два кварцевых резонатора: один на 8МГц, второй на 32768Гц, штыревой разъем для подключения отладчика. Этот плюс существенно сокращает расходы разработчика на проектирование аналогичной платы, а также позволяет моментально заняться разработкой программного обеспечения. Внешний вид микроконтроллера представлен на рисунке 4.1.1., а параметры отображены в таблице 4.1.1.

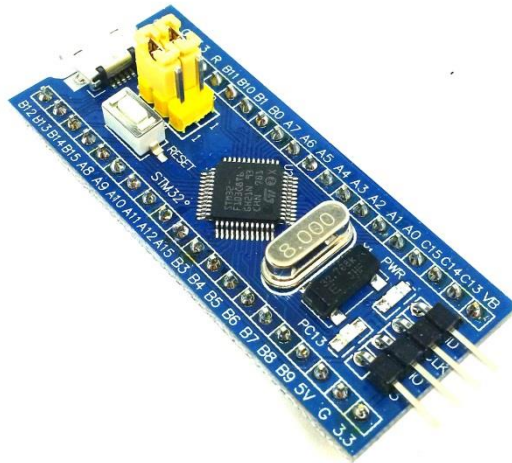


Рисунок 4.1.1 – Плата с микроконтроллером STM32F103C8T6

Таблица 4.1.1 – Параметры платы с микроконтроллером STM32F103C8T6

Параметр	Показатель
Тактовая частота, МГц	72
Размер FLASH-памяти, Кб	64
Напряжение питания, В	2.0-3.6
Размер ОЗУ, Кб	20
Максимальный ток на портах ввода-вывода, мА	25
Размеры (ДхШхВ),мм	53 x 22.5 x 6.2
Стоимость, руб.	160

### Текстово-графический LCD-дисплей

Для отображения информации применим графический LCD-дисплей на микроконтроллере ST7920. Главным фактором при выборе дисплея была его стоимость, а также возможность работы в графическом режиме. Данный дисплей может работать с применением нескольких интерфейсов: SPI,

параллельного с разрядностью шины 4 бит и параллельного с разрядностью шины 8 бит. Внешний вид дисплея представлен на рисунке 4.1.2, а параметры в таблице 4.1.2.

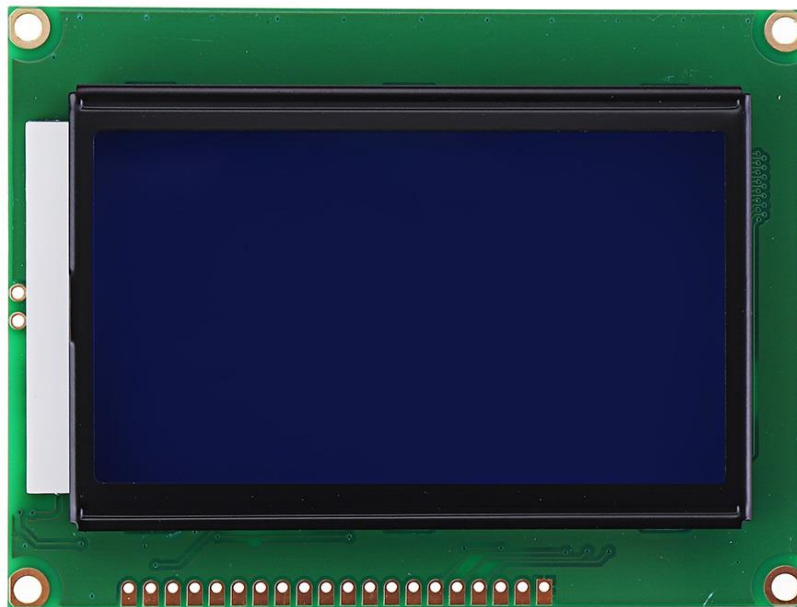


Рисунок 4.1.2 – Внешний вид LCD –дисплея

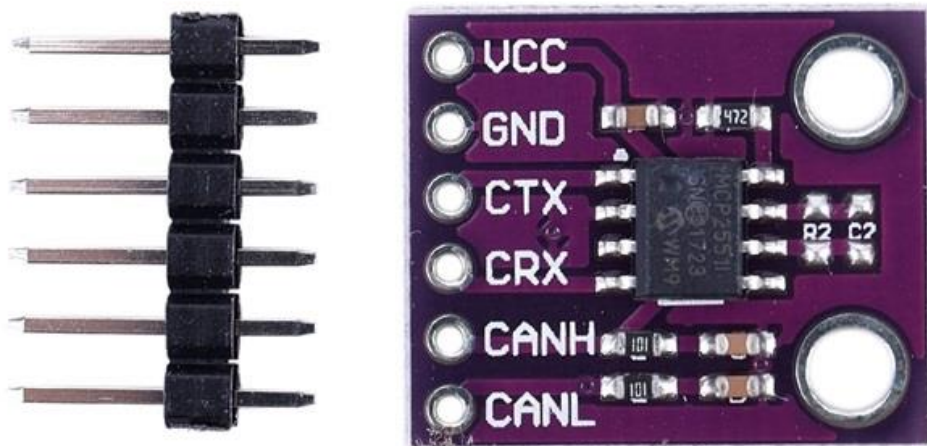
Таблица 4.1.2 – Параметры LCD - дисплея

Параметр	Показатель
Разрешение, точек	128x64
Угол обзора,град.	100 по горизонтали 60 по вертикали
Напряжение питания подсветки, В	4.9-7
Напряжение питания логических элементов, В	3.0-5.5
Рабочая температура, град. С	-20-80
Размеры (ДхШхВ),мм	120x 60 x 15
Стоимость, руб.	317



## CAN-трансивер

Для отправки и приема сообщений по CAN-шине, был выбран CAN-трансивер на микросхеме MCP2551 ввиду его невысокой стоимости, а также доступности. Внешний вид трансивера представлен на изображении 4.1.3, а параметры отображены в таблице 4.1.3.



Изображение 4.1.3 – CAN-трансивер

Таблица 4.1.3 – Параметры CAN-трансивера

Параметр	Показатель
Максимальная скорость передачи, Кбит/с	1024
Напряжение питания, В	4.5..5.5
Рабочая температура, град. С	-40..+125
Дополнительные функции	Защита от короткого замыкания, режим энергосбережения
Размеры (ДхШхВ),мм	29 x 22 x 4
Стоимость, руб.	70

## Микросхема интерфейса KWP2000

Для отправки и приема сообщений по протоколу KWP2000 была выбрана микросхема L9637D ввиду ее низкой стоимости, широкой распространенности, а также возможности работы с интерфейсом ISO-9141. Следует отметить, что в продаже не было найдено готовых решений для соединения с тестовым устройством, соответственно, была изготовлена печатная плата с данной микросхемой.

Внешний вид микросхемы представлен на изображении 4.1.4, а параметры приведены в таблице 4.1.4.



Рисунок 4.1.4 – Внешний вид микросхемы интерфейса KWP2000

Таблица 4.1.4 – параметры микросхемы интерфейса KWP2000

Параметр	Показатель
Максимальная скорость передачи, бит/с	57600
Напряжение питания, В	4.5..36
Рабочая температура, град. С	-40..+150
Дополнительные функции	Защита от короткого замыкания
Размеры (ДхШхВ), мм	15 x 20 x 4

#### Продолжение таблицы 4.1.4

Стоимость, руб.	100
-----------------	-----

#### Микросхемы интерфейса SAE J1850

Для отправки и приема сообщений по протоколу SAE J1850 VPW была выбрана микросхема AU5780, для отправки сообщений по протоколу SAE J1850 PWM была выбрана микросхема OKI M6636. Внешний вид микросхемы AU5780 представлен на изображении 4.1.5, а параметры приведены в таблице 4.1.5. Внешний вид микросхемы OKI M6636В приведен на изображении 4.1.6, а параметры приведены в таблице 4.1.6

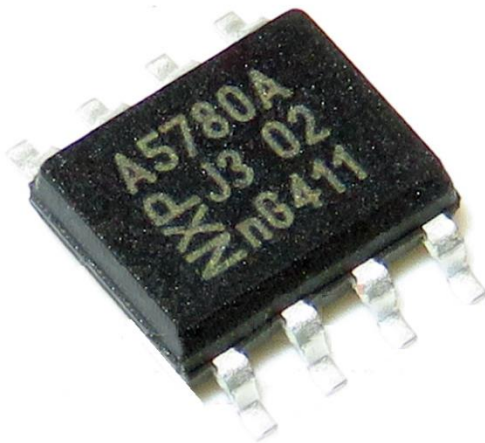


Рисунок 4.1.5 – Микросхема AU5780

Таблица 4.1.5 - параметры микросхемы AU5780

Параметр	Показатель
Максимальная скорость передачи, бит/с	10400
Напряжение питания, В	6..24
Рабочая температура, град. С	-40..+125
Дополнительные функции	Защита от короткого замыкания

Продолжение таблицы 4.1.5

Размеры (ДхШхВ), мм	5 x 4 x 2.5
Стоимость, руб.	50



Рисунок 4.1.6 – Микросхема OKI M6636B

Таблица 4.1.6 – Параметры микросхемы OKI M6636B

Параметр	Показатель
Максимальная скорость передачи, бит/с	41600
Напряжение питания, В	0.77
Рабочая температура, град. С	-50..+150
Дополнительные функции	Защита от короткого замыкания Буфер входящих сообщений
Размеры (ДхШхВ), мм	16 x 10 x 2.5
Стоимость, руб.	175

### Стабилизатор напряжения

Преобразователь напряжения необходим для снижения напряжения бортовой сети автомобиля до значения напряжения питания диагностического тестера. Так как ранее было поставлено условие, что тестер должен работать как

с бортовой сетью напряжением 12 Вольт, так и с бортовой сетью 24 Вольт, был выбран линейный стабилизатор напряжения L7805CV. Внешний вид стабилизатора представлен на изображении 4.1.7, а параметры приведены в таблице 4.1.7.



Рисунок 4.1.7 – Внешний вид стабилизатора напряжения

Таблица 4.1.7 – Параметры стабилизатора напряжения

Параметр	Показатель
Входное напряжение, В	5..35
Выходное напряжение, В	5.0±0.1
Максимальный выходной ток, А	1.5
Диапазон рабочих температур, град. С	0..125
Размеры (ДхШхВ),мм	29.6 x 10.2 x 4.5
Стоимость, руб.	21

## Штекер OBD-II

Для соединения сканера с диагностическим разъемом автомобиля потребуется штекер OBD-II, имеющий 16 контактов. Штекер был заказан с сайта «Aliexpress». Внешний вид отображен на рисунке 4.1.8.



Рисунок 4.1.8 – Внешний вид штекера OBD-II

## Программирование

Для написания программы была использована среда разработки Keil Uvision. С ее помощью можно с легкостью написать программы под любой микроконтроллер семейства STM32. К тому же, немаловажным плюсом является доступность сразу всех инструментов разработки. Недостатками ее являются отсутствие русского языка и ограничения в 32Кб кода, однако, последнее снимается при покупке полной версии.

Язык программирования в данной среде разработки – C, C++ либо Assembler. После написания кода, его сразу же можно скомпилировать и

загрузить в требуемый микроконтроллер, не прибегая к помощи сторонних программ. При отсутствии микроконтроллера всегда доступен симулятор и логический анализатор, с помощью которых можно отслеживать состояние регистров и портов ввода-вывода.

На рисунке 4.1.9 представлено окно среды разработки во время тестирования CAN-интерфейса.

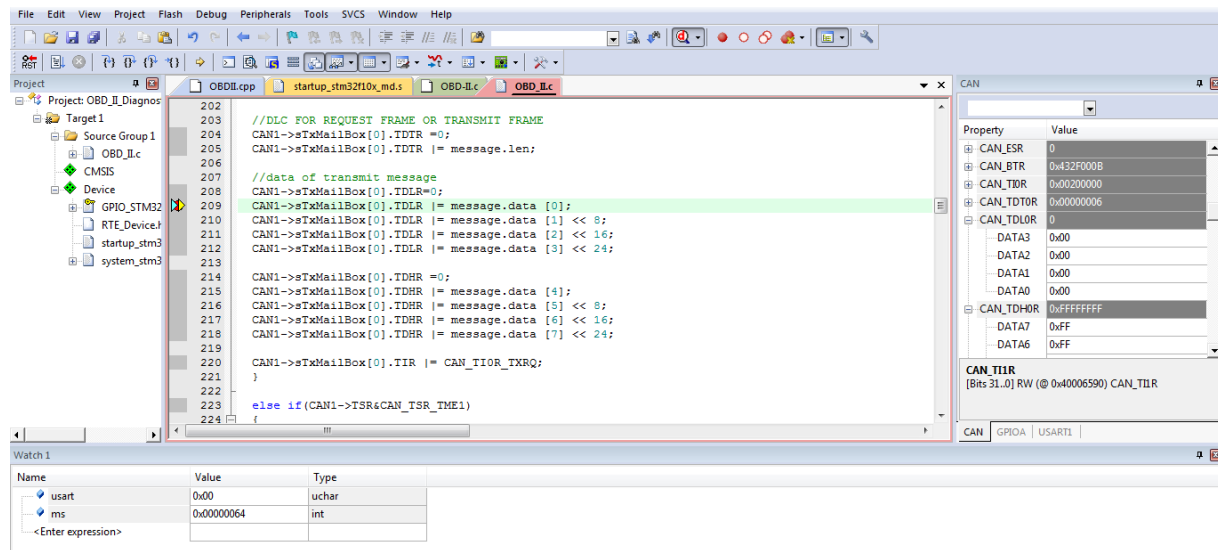


Рисунок 4.1.9 – окно среды разработки Keil Uvision

Фрагменты исходного кода программы микроконтроллера приведены в листингах В.1-В.5 приложения В.

### Сборка прототипа

Для проверки работоспособности системы был собран прототип программно-аппаратного комплекса (Рисунок 4.1.10). Кроме того, была подсчитана сумма расходов, потраченных на сборку прототипа, которая приведена в таблице 4.1.8. Следует отметить, что стоимость, указанная в таблице не является конечной, так как не были учтены затраты на провода и корпус устройства.

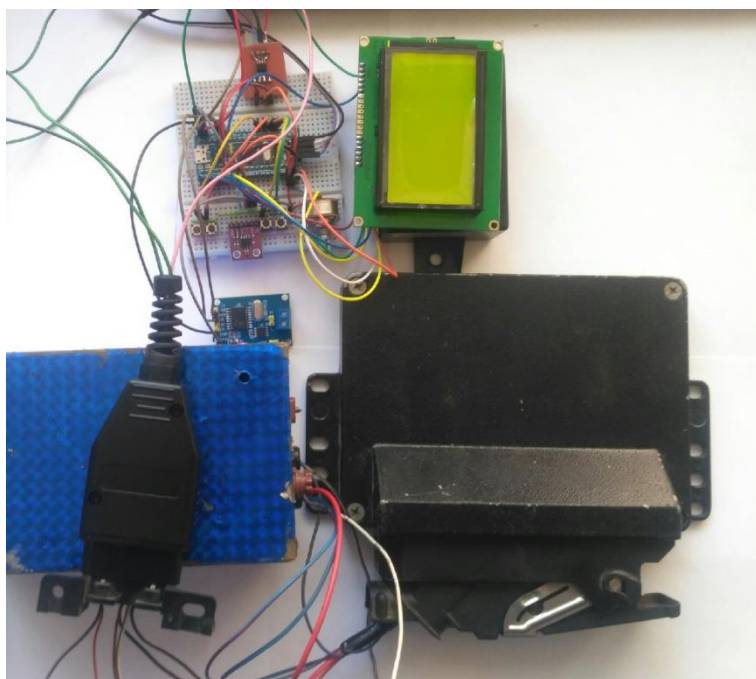


Рисунок 4.1.10 – Прототип программно-аппаратного комплекса

Таблица 4.1.8 – Экономическая составляющая программно-аппаратного комплекса

Наименование компонента	Стоимость, руб.	Количество, шт.	Сумма, руб.
Микроконтроллер STM32F103C8T6	160	1	160
LCD-дисплей	317	1	317
CAN-трансивер MCP2551	70	1	70
Микросхема интерфейса KWP2000	100	1	100
Микросхема интерфейса SAE J1850 (AU5780)	50	1	50
Микросхема интерфейса SAE J1850 (OKI M6636)	175	1	175
Стабилизатор напряжения L7805CV	21	1	21
Штекер OBD-II	218	1	218



## Продолжение таблицы 4.1.8

Тактовая кнопка	5	4	20
Итого			1131

## 5. ТЕСТИРОВАНИЕ

### 5.1. МЕТОДОЛОГИЯ ТЕСТИРОВАНИЯ

Исходя из того, что объект разработки – программно-аппаратный комплекс, был уделен большой объем внимания его программной и аппаратной частям. Производилось альфа-тестирование имеющегося функционала.

Альфа тестирование – один из видов тестирования, в котором проверка работоспособности и соответствия требованиям объекта недоступна конечному пользователю, а доступна разработчику и / или потенциальному заказчику. Альфа-тестирование может производиться как на ранней, так и на поздней стадиях разработки продукта, что является несомненным плюсом, так как у тестировщика появляется возможность искать ответы на возникшие проблемы по мере их поступления и видеть причинно-следственную связь при возникновении этих проблем.

Процедура тестирования производилась с применением электронного блока «Январь 5.1-1411020-41». Применяемая в этом ЭБУ линия связи – К-линия (протокол KWP2000), загруженное программное обеспечение – J5LS\_v43.

### 5.2. ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ

Проведем тестирование автоопределения присутствующих линий связи. Включим программно-аппаратный комплекс, разорвем линию связи путем отсоединения провода на макете. Результат работы представлен на рисунке 5.2.1



Рисунок.5.2.1 – Результат процедуры автоопределения

Тест показал, что линия не определена. Протестируем возможность вручную выбрать линию связи CAN BUS и J1850, чтобы убедиться, то кнопки навигации программно-аппаратного комплекса работают корректно. Результаты отображены на рисунках 5.2.2 и 5.2.3.



Рисунок 5.2.2 – Результат выбора линии связи CAN BUS



Рисунок 5.2.3 - Результат выбора линии связи J1850

Тест показал, что кнопки навигации подключены корректно и исправно работают.

Проведем попытку соединения с электронным блоком по К-линии, заодно проверим навигацию по основному меню. Результат представлен на рисунке 5.2.4 и рисунке 5.2.5.

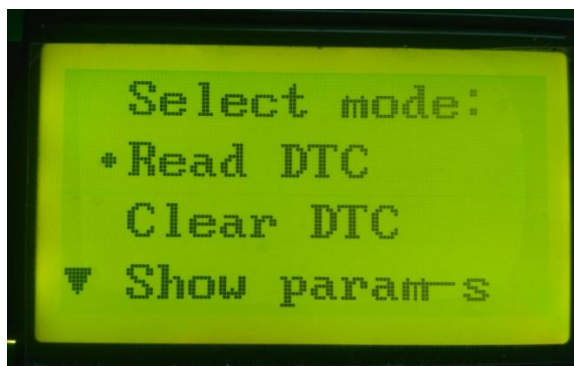


Рисунок 5.2.4 – Результат соединения с электронным блоком

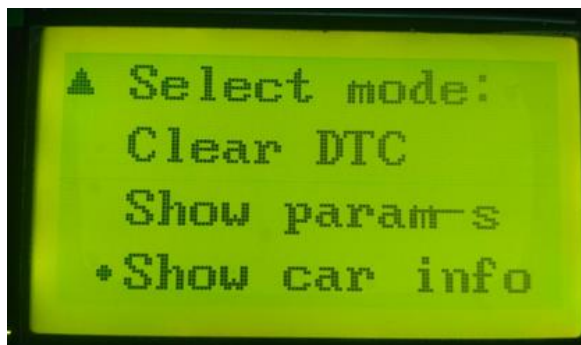


Рисунок 5.2.5 – результат навигации по основному меню программы

Произведем тестирование обмена данными с электронным блоком в режиме реального времени, отображение характеристик ЭСУД, чтение и расшифровку диагностических кодов неисправностей. Результаты отображены на рисунках 5.2.6-5.2.9.



Рисунок 5.2.6. – Отображение параметров ЭСУД в реальном времени

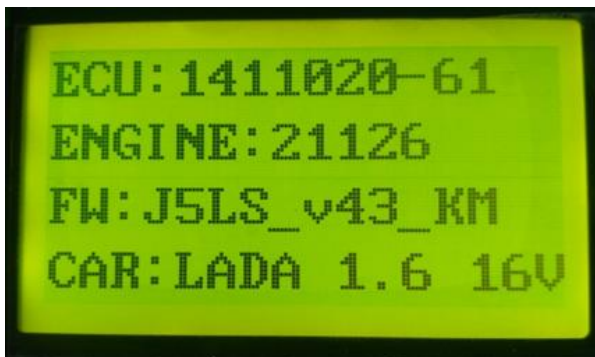


Рисунок 5.2.7. – Отображение характеристик ЭСУД

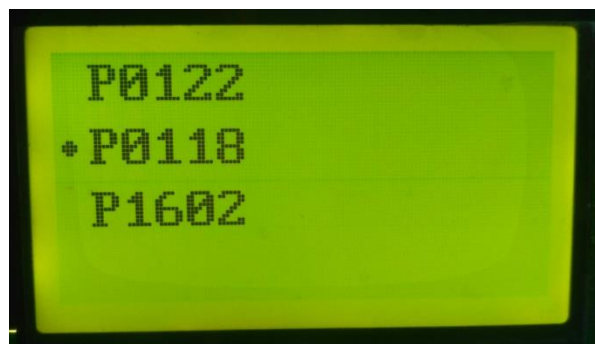


Рисунок 5.2.8. – Отображение имеющихся кодов неисправностей в памяти  
ЭСУД

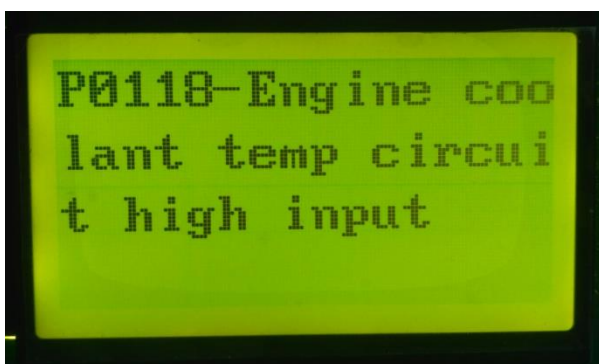


Рисунок 5.2.9. – Расшифровка кода неисправности ЭСУД

Как видно из результатов тестирования, основные функции программно-аппаратного комплекса выполняются, идет обмен данными с электронным блоком автомобиля, производится чтение кодов неисправностей, параметров системы управления двигателем и пр., значит испытание макета проведено успешно.

## 6. ЗАКЛЮЧЕНИЕ

В представленной работе были проработаны все этапы, связанные с проектированием, разработкой и вводом в эксплуатацию программно-аппаратного комплекса,

В результате обзора аналогов обнаружилось, что большинство из них имеют завышенную стоимость, а также зависимость от питания. Не все из них умеют автоматически определять присутствующую в автомобиле линию связи. Эти недостатки были учтены во время проектирования.

Были рассмотрены три способа коммуникации диагностического оборудования с электронными блоками, а именно: протокол J1850, протокол KWP2000, протокол CAN 2.0A. Также были рассмотрены режимы работы диагностического оборудования.

Был разработан и введен в эксплуатацию программно-аппаратный комплекс, позволяющий производить диагностику транспортного средства и наблюдение за параметрами системы управления двигателем в режиме реального времени. Следует отметить, что была проведена стоимостная оценка проекта. Стоимость готового решения получилась в несколько раз ниже аналогов.

Производилось альфа-тестирование проекта. Было проведено тестирование с использованием протокола KWP2000. Программно-аппаратный комплекс получил возможность автоматически определять присутствующие линии связи в автомобиле, отображать параметры ЭСУД в режиме реального времени, управлять исполнительными механизмами.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Hartmann, J. How to Tune and Modify Automotive Engine Management Systems / J. Hartmann. – Minneapolis: Motorbooks, 2013. – 333 p.
2. Веб-сайт «Chiptuner». [Электронный ресурс]. URL: <https://chiptuner.ru/>. Дата обращения – февраль 2020 года.
3. Веб-сайт «Launchrus». [Электронный ресурс]. URL: <https://www.launchrus.ru/products/multibrand-scanners/pilot-scan/>. Дата обращения – февраль 2020 года.
4. Веб-сайт «Autel-Russia». [Электронный ресурс]. URL: <https://autel-russia.ru/catalog/skanery-serii-maxisys/maxisys-ms906.html>. Дата обращения – февраль 2020 года.
5. Веб-сайт «Launchrus». [Электронный ресурс]. URL: <https://www.launchrus.ru/products/multibrand-scanners/sniatye-s-proizvodstva/x-431-pro/>. Дата обращения – февраль 2020 года.
6. Веб-сайт «Autoscanners». [Электронный ресурс]. URL: [https://www.autoscanners.ru/catalogue/?catalogue\\_id=izmeritelnoe\\_oborudovanie](https://www.autoscanners.ru/catalogue/?catalogue_id=izmeritelnoe_oborudovanie). Дата обращения – февраль 2020 года.
7. Gunther, M. Ignition Systems for Gasoline Engines / M. Gunther. – Luxembourg City: Springer International Publishing, 2017. – 331 p.
8. Boys, R. CAN Primer : Creating your own Network / R. Boys. – San Jose: Keil, 2009. – 13p.
9. ГОСТ Р ИСО 11898-1-2015. ТД. Местная контроллерная сеть (CAN). Канальный уровень и передача сигналов. – М.:Стандартинформ, 2016. – 42с.
10. Robert Bosch GmbH Can Specification / Robert Bosch GmbH. – Stuttgart: Robert Bosch GmbH, 1997. – 73p.
11. ISO 14230-1. RV-DS Keyword Protocol 2000 part 1: Physical layer. – Switzerland.: International Organization for Standardization, 1999. – 12p.

- 12.ISO 14230-2. RV-DS Keyword Protocol 2000 part 2: Data link layer. – Switzerland.: International Organization for Standardization, 1999. – 34p.
- 13.ISO 14230-3. RV-DS Keyword Protocol 2000 part 3: Application layer. – Switzerland.: International Organization for Standardization, 2000. – 139p.
- 14.ISO 14230-4. RV-DS Keyword Protocol 2000 part 4: Requirements for emission-related systems. – Switzerland.: International Organization for Standardization, 2000. – 15p.
- 15.SAE J1850. Class B Data Communication Network Interface. – Warrendale: Society of Automotive Engineers, 2001. – 24p.
- 16.Oliver, J. Implementing the J1850 Protocol / J. Oliver. - Warrendale: Society of Automotive Engineers, 2001. – 15p.
- 17.Веб-сайт «Microchip». [Электронный ресурс]. URL: <https://www.microchip.com/design-centers/8-bit/avr-mcus>. Дата обращения – март 2020 года.
- 18.Веб-сайт «Microchip». [Электронный ресурс]. URL: <https://www.microchip.com/design-centers/8-bit/pic-mcus>. Дата обращения – март 2020 года.
- 19.Веб-сайт «ST Microelectronics». [Электронный ресурс]. URL: <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html>. Дата обращения – март 2020 года.
- 20.Веб-сайт «Riptutorial». [Электронный ресурс]. URL: <https://riptutorial.com/ru/stm32/topic/7741/интегрированные-среды-разработки--ide->. Дата обращения – март 2020 года.
- 21.Веб-сайт «Schemnet». [Электронный ресурс]. URL: <https://schem.net/software/keil.php>. Дата обращения – март 2020 года.
- 22.Хоровиц, П. Искусство схемотехники / П. Хоровиц, У. Хилл – М.: Издательство БИНОМ, 2014. – 704с.



- 23.Веб-сайт «Arm Developer». [Электронный ресурс]. URL: <https://developer.arm.com/downloads/cmsis>. Дата обращения – март 2020 года.
- 24.Веб-сайт «ST Microelectronics». [Электронный ресурс]. URL: <https://www.st.com/en/embedded-software/stm32-standard-peripheral-libraries.html>. Дата обращения – март 2020 года.
- 25.Веб-сайт «Themagicsmoke». [Электронный ресурс]. URL: <https://themagicsmoke.ru/courses/stm32/hal.html>. Дата обращения – март 2020 года.
- 26.Веб-сайт «Compel». [Электронный ресурс]. URL: <https://www.compel.ru/lib/88223>. Дата обращения – март 2020 года.
- 27.SAE J1978. OBD II Scan tool. – Warrendale: Society of Automotive Engineers, 2001. – 7р.
- 28.SAE J1979. SVS E/E Diagnostic Test Modes. – Warrendale: Society of Automotive Engineers, 2002. – 159р.
- 29.SAE J2190. SVS Enhanced E/E Diagnostic Test Modes. – Warrendale: Society of Automotive Engineers, 2008. – 62р.
- 30.SAE J2178. SVRP Class B Data Communication Network Messages. – Warrendale: Society of Automotive Engineers, 2008. – 62р.
- 31.Веб-сайт «Чип и Дип». [Электронный ресурс]. URL: <https://www.chipdip.ru/>. Дата обращения – апрель 2020 года.
- 32.Веб-сайт «ПРОКОНТАКТ. Все для радиотехника». [Электронный ресурс]. URL: <https://procontact74.ru/>. Дата обращения – апрель 2020 года.
- 33.Веб-сайт «Промэлектроника». [Электронный ресурс]. URL: <https://www.promelec.ru/>. Дата обращения – апрель 2020 года.
- 34.Веб-сайт «Aliexpress». [Электронный ресурс]. URL: <https://aliexpress.ru/>. Дата обращения – апрель 2020 года.