

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

_____ 2023 г.
«__» _____

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ЭВМ
_____ Д. В. Топольский
«__» _____ 2023 г.

Разработка системы управления электротрансмиссией для гибридного
и электротранспорта

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУРГУ-090401.2023.469 ПЗ ВКР

Консультант,
Доцент каф. ЭВМ
_____ П. О. Шабуров
«__» _____ 2023 г.

Руководитель работы,
д.т.н., профессор каф. ЭВМ
_____ С. А. Ганджа
«__» _____ 2023 г.

Автор работы,
студент группы КЭ-222
_____ В. В. Кондратенко
«__» _____ 2023 г.

Нормоконтролёр,
ст. преп. каф. ЭВМ
_____ С. В. Сяськов
«__» _____ 2023 г.

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

_____ Д. В. Топольский

« ___ » _____ 2023 г.

ЗАДАНИЕ

на выпускную квалификационную работу магистра

студенту группы КЭ-222

Кондратенко Виктору Витальевичу,

обучающемуся по направлению

09.04.01 «Информатика и вычислительная техника»

1. **Тема работы:** «Разработка системы управления электротрансмиссией для гибридного и электротранспорта» утверждена ректора 25 апреля 2023г №753-13/12, приложение №308/10.
2. **Срок сдачи студентом законченной работы:** 1 июня 2023 г.
3. **Исходные данные к работе:**
 - Мотор-колесо на основе вентильного электродвигателя комбинированного возбуждения;
 - номинальная мощность 30 кВт;
 - номинальное напряжение питания 72 В;
 - максимальный допустимый ток на цепь якоря 1000 А;
 - максимальный допустимый ток на цепь возбуждения 100 А;
 - максимальные обороты 16000 об/мин на редукторе и до 1600 об/мин на колесо;

- электронные компоненты системы управления;
- коммутатор управления;
- протокол передачи данных CAN-шина;
- микроконтроллер на базе STM32.

4. Перечень подлежащих разработке вопросов:

1. Анализ предметной области;
2. Определение требований системы;
3. Подбор компонентной базы;
4. Реализация управления и схемы подключения.

5. Дата выдачи задания: 02.12.2022 г.

Руководитель работы _____ /С. А. Ганджа/

Студент _____ /В. В. Кондратенко/

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Анализ предметной области	10.03.2023	
Определение требований системы	21.03.2023	
Подбор компонентной базы	04.04.2023	
Реализация управления и схемы подключения	25.04.2023	
Компоновка текста работы и сдача на нормоконтроль	16.05.2023	
Подготовка презентации и доклада	24.05.2023	

Руководитель работы _____ /С. А. Ганджа/

Студент _____ /В. В. Кондратенко/

АННОТАЦИЯ

В. В. Кондратенко. Разработка системы управления электротрансмиссией для гибридного и электротранспорта. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2023, 52 с., 24 ил., библиогр. список – 26 наим.

В ходе выполнения выпускной квалификационной работы проведено ознакомление с трансмиссией на основе вентильного электродвигателя комбинированного возбуждения. Рассмотрен алгоритм управления данным мотор-колесом, произведен подбор компонентов управления, подходящие для условий работы, спроектирована схема подключения системы управления электротрансмиссией, реализована эмуляция управления педалью, а также получение данных с микроконтроллера.

Данная работа соответствует приоритетным направлениям научно-технического развития страны, включающее продолжением развития теории вентильных машин комбинированного в части управления и анализа технического состояния.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	7
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	9
1.1 Анализ технологического решения.....	9
1.2 Анализ алгоритма управления.....	11
2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ СИСТЕМЫ.....	16
3 ПОДБОР КОМПОНЕНТНОЙ БАЗЫ.....	22
4 РЕАЛИЗАЦИЯ УПРАВЛЕНИЯ И СХЕМЫ ПОДКЛЮЧЕНИЯ	31
4.1 Элементы системы управления	31
4.2 Алгоритм системы управления	38
4.3 Система мониторинга показателей	42
ЗАКЛЮЧЕНИЕ	43
БИБЛИОГРАФИЧЕСКОЙ СПИСОК	44
ПРИЛОЖЕНИЯ.....	46
Приложение А (Исходный код скриптов для эмуляции).....	46
Приложение Б (Исходный код скриптов для обмена данными).....	51

ВВЕДЕНИЕ

Технологии развиваются и стремятся быть высокопроизводительными, поэтому происходит активный переход на использование гибридного и электротранспорта. Основными преимуществами использования такой трансмиссии уменьшает создаваемый шум, что является проблемой в больших городах, выделяет меньше или не выделяет вредные газы при своей работе, а также потребляем меньше “топлива”.

Преимущества гибридного привода заключаются, прежде всего, в низком расходе топлива, а также в том, что электродвигатель смягчает все неблагоприятные рабочие диапазоны ДВС. Кроме того, мощностные характеристики используемого электродвигателя и ДВС прекрасно сочетаются, так как низкий крутящий момент двигателя внутреннего сгорания идеально дополняется более высоким крутящим моментом электродвигателя (рисунок 1). Другие преимущества заключаются в отсутствии стартера и генератора (в зависимости от выбранной гибридизации), так как их функции в этом случае выполняют электродвигатели. Кроме того, регенерация энергии торможения может минимизировать износ тормозных механизмов [1].

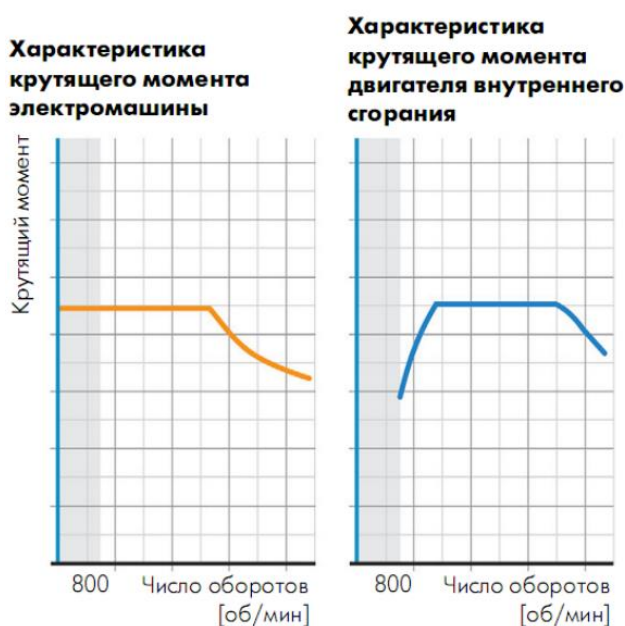


Рисунок 1 - Графики характеристик крутящего момента электродвигателя и ДВС

Достоинством электромеханических трансмиссий является обеспечение наиболее широкого диапазона автоматического изменения крутящего момента и силы тяги, отсутствие жёсткой кинематической связи между агрегатами электромеханической трансмиссии, что позволяет создать различные компоновочные схемы [2].

Целью, представленной выпускной квалификационной работы, является разработка части системы управления, описание схемы подключения компонентов управления, перенос эмуляции управления на микроконтроллер с джойстиком с использованием STM32CubeIDE, а также реализация получения данных по USB-интерфейсу с использованием PySerial.

Для достижения поставленной цели, необходимо решить следующие поставленные задачи:

- ознакомление с текущей реализацией управления;
- подбор компонентов для модернизации системы управления;
- подбор микроконтроллера для управления всей системой управления;
- описание теоретической схемы подключения системы управления к электротрансмиссии;
- адаптация программного кода для микроконтроллера по эмуляции управления с помощью джойстика, с использованием STM32CubeIDE;
- реализация программного кода получения информации с USB-интерфейса микроконтроллера на компьютер с использованием PySerial;
- проведение исследовательских испытаний.

Так как данную разработку можно реализовать разными путями, то результаты, удовлетворяющие как разработчика, так и независимых экспертов, будут достигаться постепенно, используя эмпирические методы.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Анализ технологического решения

Один из вариантов располагает к наибольшей унификации с автомобилем с механической трансмиссией. Он заключается в передаче крутящего усилия с электродвигателя на карданные валы без использования коробки передач. При этом кинематика модернизированного автомобиля остается такой же, как и у базового автомобиля. Достигается более высокий показатель экологичности и плавности хода, старта. Ресурс двигателя внутреннего сгорания повышается. Из минусов данного решения компоновки электромеханической трансмиссии можно выявить снижение КПД трансмиссии, увеличение общей массы системы, а также добавление силовой электроники, требующей микропроцессорного управления.

Второй вариант компоновки электромеханической трансмиссии позволяет уменьшить количество механических частей трансмиссии, но это сказывается на унификации с базовым автомобилем. В данной конструкции появляется возможность подключения электродвигателей по мере необходимости и в зависимости от дорожных условий, что положительно сказывается на ресурсе, экологичности и проходимости модернизированного автомобиля. Возникает проблема программной синхронизации электродвигателей при данной компоновке трансмиссии. Поэтому необходима установка дополнительных контроллеров и датчиков, а также систем контроля движения (ABS, ESP).

Наиболее перспективным вариантом электромеханической трансмиссии является использование мотор-колес. При данном варианте можно максимально уйти от механической части трансмиссии. При применении данного вида компоновки электромеханической трансмиссии получаем независимую подвеску, центр тяжести автомобиля смещается ближе к дорожному полотну. Появляется возможность взаимозаменяемости мотор-колес и рекуперации электроэнергии. Основными критериями,

определяющими целесообразность применения индивидуального электропривода с мотор-колесами на транспортных средствах вместо механического, являются следующие:

- вес агрегатов и узлов привода в сравниваемых вариантах;
- эксплуатационные свойства (надежность в работе, объем и периодичность технического обслуживания);
- КПД системы привода;
- срок службы агрегатов;
- стоимость агрегатов и узлов привода в сравниваемых вариантах и эксплуатационные расходы за расчетный срок службы.

Характерной особенностью конструкции мотор-колеса является весьма рациональная компоновка его элементов, в частности, размещение электродвигателя полностью или частично внутри обода колеса, а также сведение к минимуму числа и размеров деталей механической передачи.

В данной научной работе реализация системы управления будет вестись с конкретным мотор-колесом. Данная конструкция была запатентована как вентильный электродвигатель комбинированного возбуждения в Южно-Уральском государственном университете. Данная конструкция обладает преимуществами электрических машин комбинированного возбуждения и при этом нивелирует недостатки других конструкций. Эта конструкция может являться основой для реализации различного ряда мотор-колес под разные нужды, то есть имеется вариативность мощности и частоте вращения.

Рассматриваемая вентильная машина обладает постоянными магнитами, что означает неизменность магнитного потока. Управление частотой вращения вентильного двигателя, а, следовательно, его моментом (Нм) может происходить по цепи якоря и цепи возбуждения, данная возможность позволяет иметь широкий диапазон регулирования. При этом регулировка по цепи возбуждения является наиболее выгодной, так как по ней проходит примерно от 1-5 % потребляемой мощности двигателя (рисунок 2).



Рисунок 2 – Компоновка трансмиссии мотор-колеса

Регулирование крутящего момента и частоты вращения в ВЭЖВ происходит как за счет регулирования величины питающего напряжения, так и за счет регулирования магнитного потока, создаваемого обмоткой возбуждения. Данный принцип позволяет обеспечить широкий диапазон регулирования скорости и момента при относительно малой массе и габаритах. Использование высококоэрцитивных постоянных магнитов позволяет сэкономить массу и объем двигателя, что позволяет разместить его непосредственно в колесе электромобиля.

1.2 Анализ алгоритма управления

На начальном этапе считывается вся информация, необходимая для расчета статических и динамических моментов трансмиссии.

Далее алгоритм переходит к определению параметров и характеристик электродвигателя. Первоначально определяется результирующий магнитный поток электродвигателя, который должен обеспечить баланс между напряжением источника питания, ЭДС вращения и падением напряжения в якорной цепи. Поскольку магнитный поток от постоянных магнитов остается

неизменным, этот баланс обеспечивается током обмотки возбуждения и, соответственно, магнитным потоком от обмотки возбуждения.

Баланс напряжений будет в любой ситуации достигнут, так как ток обмотки возбуждения может менять знак, и за счет этого суммарный магнитный поток можно теоретически уменьшить до нуля.

При низкой частоте вращения этот баланс можно достичь за счет снижения напряжения питания электродвигателя практически до нуля.

Далее по алгоритму определяется ток якоря, для создания необходимого момента для преодоления суммарного статического и динамического моментов сопротивления. Этот ток заведомо меньше тока короткого замыкания и может быть достигнут, если в системе нет токоограничения по цепи якоря. Если требуемый по балансу моментов якорный ток максимально допустимый ток, то алгоритм переходит в режим токоограничения. При этом ток якоря фиксируется на уровне максимально допустимого по условиям ограничения тока якоря. Пересчитывается максимально возможный момент, который в этих условиях может создать электродвигатель, определяется новое значение динамического момента, пересчитывается частота вращения электродвигателя и скорость движения транспортного средства. Понятно, что в этой ситуации транспортное средство будет двигаться с меньшей скоростью, которая запланирована. Новые параметры движения фиксируются. Алгоритм переходит на заключительный этап работы.

В заключительном блоке алгоритма определяются электрические (в обмотке якоря и обмотке возбуждения) потери, магнитные потери в зубцах якоря и в спинке якоря, добавочные и механические потери. При этом учитывается изменение потока от обмотки возбуждения при изменении частоты вращения электродвигателя. Параметры элементарных участков суммируются в интегральные характеристики и параметры электродвигателя и электротрансмиссии за счет циклов, суммирующих показатели элементарных участков.

В алгоритме (рисунок 3) предусмотрено измерение следующих 15 динамических значений электротрансмиссии в целом и конкретно ВЭКВ:

- проходимая транспортным средством дистанция;
- скорость транспортного средства;
- ускорение транспортного средства;
- ток возбуждения ВЭКВ;
- ток якоря ВККВ;
- напряжение питания ВЭКВ;
- момент на валу ВЭКВЖ;
- частота вращения ВЭКВ;
- температура нагрева ВЭКВ;
- суммарный момент электротрансмиссии;
- потребляемая электротрансмиссией мощность;
- мощность на валу ВЭКВ;
- общие потери ВЭКВ;
- затрачиваемая электроэнергия электротрансмиссией при движении по трассе;
- затрачиваемая АКБ емкость в А*ч с учетом рекуперации.

- получение информации о положении джойстика, перевод положения в коэффициент ускорения на основе требуемой скорости в дистанции;
- опрос текущих показателей с датчиков на ОВ и коммутатора;
- если ускорение изменилось. Расчёт необходимых показателей тока для обмотки возбуждения (ОВ) и обмотки якоря (ОЯ). Исходя из алгоритма просчета показателей, при старте, то есть начальном разгоне – ток на ОВ стремится к I_{max} . После ток на ОЯ стремится к I_{max} . Если необходимо большее ускорение, то ток на ОЯ = I_{max} , а ток на ОВ падает;
- отправка информационного сигнала по шине к коммутатору о необходимых показателях. Также включение силового ключа через оптопару на ОВ;
- повторение пункта 2;
- при достижении необходимого тока на ОВ. Генерация функции ШИМ-сигнала, которым будет вестись управление оптопарой по циклам открытия/закрытия силового ключа, тем самым поддержанием необходимого тока;
- возврат к пункту 1.

Особенность работы вентильных двигателей заключается в том, что бесконтактный токоподвод к нему осуществляется через электронный коммутатор.

2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ СИСТЕМЫ

Теоретически для вентильного двигателя можно применить векторное управление и дискретную коммутацию.

Векторное управление имеет сложный алгоритм. Достоинством его является плавное управление вектором намагничивающей силы поля якоря. По энергетическим характеристикам этот тип коммутации превосходит другие известные типы, но этот способ управления является самым дорогим. Его имеет смысл применять для высокоточных приводов. Для электротрансмиссии транспортного средства это достаточно дорого, хотя достаточно многочисленные попытки развивать колесо в этом направлении имеются.

Двигатель имеет комбинированное возбуждение, то есть происходит управление вентильной частью (рисунок 4) и обмоткой возбуждения (рисунок 5)

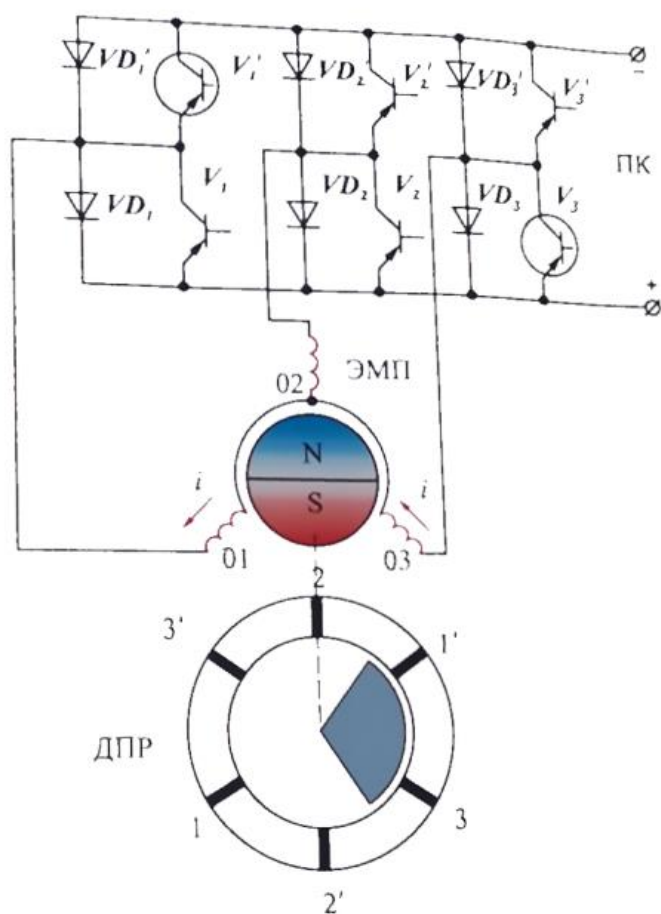


Рисунок 4 - Принципиальная электрическая схема вентильного двигателя

Схема управление двигателя с использованием 120-градусной дискретной коммутации, а для управления схемой с обмоткой возбуждения спроектирована следующая схема.

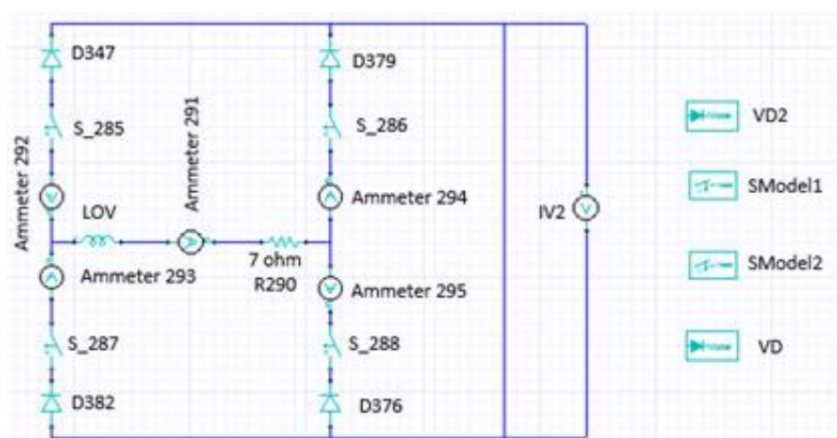


Рисунок 5 – Схема обмотки возбуждения

Мотор-колесо уже имеет свой коммутатор для управления током на цепи якоря, но нет управления током на цепи возбуждения, что и является задачей по проектированию и реализации управления тока возбуждения и коммутатора. С помощью использования микроконтроллера и дополнительных электронных компонентов с датчиками можно спроектировать схему управления, которая будет управляться через ШИМ сигнал, задавая необходимую частоту вращения колеса.

Существует множество микроконтроллеров на различных процессорах, такие как Arduino, Raspberry Pi, STM32 и другие. В данной задаче важна скорость просчёта необходимого тока для подачи, формирование информационного сигнала, обработка информационного сигнала, наличие необходимых выходов, удобство работы, стоимость. Для данной работы рассматриваются указанные ранее микроконтроллеры:

- *Arduino* является неплохим вариантом для начинающих, но также и подходит для сборки сложных схем, данный микроконтроллер является 8-битным, что может пагубно себя повести в случае, когда необходимы быстрые и сложные операции для управления сложными системами с обратными связями [4];

- *Raspberry Pi* уже является по большей части микрокомпьютером, который может иметь установленную операционную систему и подключать несколько мониторов с большим разрешением, данный микроконтроллер обладает излишней мощностью, а также обладает высокой стоимостью, что может повлиять на массовое производство электро-трансмиссий на базе управления данным микроконтроллером [5];
- *STM32* чипы в рамках каждой из которых используется одно и то же 32-битное ядро ARM. Каждый микроконтроллер состоит из ядра процессора, статической RAM-памяти, флеш-памяти, отладочного и различных периферийных интерфейсов. Данный микроконтроллер является таким же бюджетным устройством, как и Arduino, но имеет уже более производительное ядро, что выгодно для нас [6].

На основе вышеперечисленных микроконтроллерах выбор падает на использование *STM32*. Так как данный микроконтроллер является не отечественной разработкой, то его доступность может быть под вопросом, но существуют различные аналоги данного микроконтроллера, такие как *GigaDevice* китайского производства, которая имеет совместимые по распиновке и параметрам периферии с линейкой микроконтроллеров *STM32* [7].

Управление мотор-колесом как говорилось ранее можно разбить на две части, управление током якоря и током цепи возбуждения, ускорение, на основе которого будет определять необходимый подаваемый ток рассчитывается с помощью ШИМ-контроллера, которым может являться джойстик. Преимущество использования джойстика позволяет без дробления на педаль и рулевое колесо управлять мощностью колес на электро-трансмиссии.

Для управления током якоря необходимо генерировать информационный сигнал, который будет отправляться по шине к

информационному порту на коммутаторе, следовательно, будет реализовано прямое подключение через интерфейс передачи данных микроконтроллер-коммутатор, для этого необходим порты на микроконтроллере. Для регулировки также учитывается обратная связь в виде текущей скорости, которое обрело мотор-колесо, значение скорости микроконтроллер получает по шине обратно, проводя перерасчет и корректирую информационный сигнал отправляет его обратно.

Рассматриваются следующие интерфейсы для передачи данных по шине, RS-485 и CAN. CAN и RS-485 похожи тем, что оба они используют дифференциальную сигнализацию, это дает им обоим хорошую устойчивость к синфазным помехам. RS-485 был определен для промышленного рынка, в то время как CAN был в первую очередь разработан для транспортного сегмента. RS-485 нет стандартного способа решить, кто должен отправлять, что отправляется, как узнать, что оно дошло неповрежденным и т. д. CAN определяет полные пакеты на шине, которые включают 16-битную контрольную сумму CRC [8].

Еще одно важное различие между CAN и RS-485, о котором уже упоминалось, заключается в том, что RS-485 активно приводится в оба состояния, в то время как CAN всегда приводится только в доминирующее состояние, а сама шина переходит в рецессивное состояние, это имеет существенное значение на более высоких уровнях протокола для арбитража шины. Исходя из данного сравнение и при присутствии CAN-шины на управляющем коммутаторе для контролирования тока якоря, будет использоваться CAN-шина [9].

Для управления током на цепи возбуждения необходимо подавать сигналы определенной длины, генерируемы автоматической функцией ШИМ сигнала через выходы таймера микроконтроллера, на вход оптопары, которая в свою очередь будет замыкать и размыкать силовой ключ питания на обмотку возбуждения, в данном случае оптопара выполняет роль драйвера по управлению мощностью подаваемого тока.

Для генерации ШИМ-сигнала с заданной скважностью будет использовано 4 вывода таймера STM32, для этого необходимо настроить соответствующий таймер и выбрать каналы для генерации ШИМ-сигнала. необходимо выполнить следующие шаги:

- выбрать таймер и каналы для генерации ШИМ-сигнала;
- настроить период ШИМ-сигнала (Auto-Reload Register, ARR);
- настроить частоту работы таймера путем деления системной частоты на значение предделителя +1 (Prescaler, PSC);
- выбрать режим работы каналов таймера на генерацию ШИМ-сигнала (Capture/Compare Mode Register, CCMR);
- включить выходные сравнения для выбранных каналов таймера (Capture/Compare Enable Register, CCER);
- установить значение скважности для каждого канала. Для каждого выбранного канала необходимо установить значение скважности с помощью соответствующего регистра захвата/сравнения (CCR);
- значение скважности определяет отношение времени импульса к периоду сигнала;
- запустить таймер. После настройки всех параметров таймера и выбранных каналов, запустить таймер с помощью соответствующего регистра управления (CR1) [10].

Для управления цепью возбуждения также необходимы дополнительные электронные компоненты. Чтобы реализовать обратную связь и корректно управлять током на обмотке возбуждения необходим датчик тока, которым может являться датчиками компании Allegro. Максимальный подаваемый ток значительно меньше, чем на цепи якоря, и выбор датчика тока до 100 А будет с запасом. Следовательно, также подбирается и необходимый нам силовой транзистор на 100 А с запасом.

Для мониторинга состояния мотор-колеса и сбора показателей, таких как текущий момент, необходимо подключение микроконтроллера к другому устройству, такому как компьютер. Для этого используется USB CDC на

микроконтроллере с переходником на mini-USB, при подключении к компьютеру будет создан виртуальный COM-порт, через который можно получать данные, сохранять их и представлять в виде графиков с использованием дополнительного программного обеспечения.

3 ПОДБОР КОМПОНЕНТНОЙ БАЗЫ

Возьмем за базу отладочную плату STM32F407 Discovery (рисунок 6). Данная плата имеет требуемое количество необходимых для работы пинов подключения, также для последующей модернизации или изменения подхода к управлению имеется запас [11].

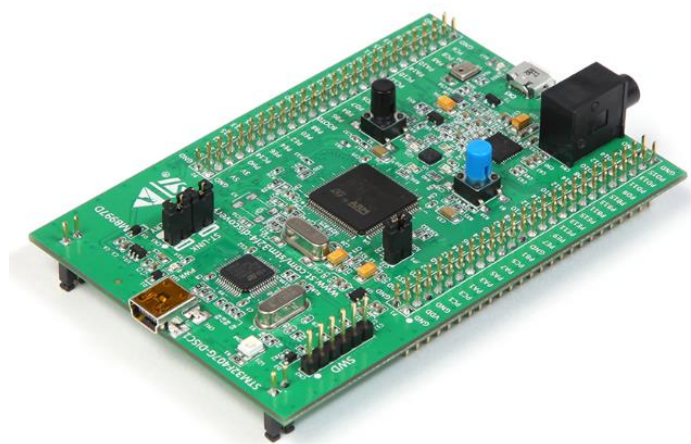


Рисунок 6 - STM32F407 Discovery

В связи с недоступностью данного микроконтроллера в большом количестве на территории страны можно использовать аналог STM контроллеров от компании GigaDevice. Например для данного МК это является GD32450V (рисунок 7) [12].



Рисунок 7 - GD32450V-Start GigaDevice

При переносе кода с микроконтроллера STMicro на микроконтроллер Gigadevice усилия по портированию могут повлиять многие факторы, такие как использование библиотек периферии, использование доступа к регистрам на уровне регистров и использование генератора кода. Если используются библиотеки периферии, усилия по портированию минимальны. Однако при использовании доступа к регистрам на уровне регистров каждый отдельный регистр должен быть проверен, что является самой сложной частью портирования для генераторов кода [13].

Используемые выводы таймеров

Данный МК имеет двенадцать универсальных 16-битных таймеров, включая два ШИМ-таймера для управления двигателем, два 32-битных таймера общего назначения. А также настоящий генератор случайных чисел (ГСЧ). Они также имеют стандартные и расширенные коммуникационные интерфейсы.

Имеются 12 универсальных таймеров (General-purpose timers), они могут быть настроены для работы в различных режимах, таких как счетчик времени, ШИМ-генератор, захват входного сигнала и другие. Данные таймеры обладают 4-мя независимыми каналами.

Также имеются таймеры с расширенным управлением (TIM1, TIM8) можно рассматривать как трехфазные генераторы ШИМ, мультиплексированные по 6 каналам. Они имеют дополнительные ШИМ-выходы с программируемыми вставленными мертвыми временами. Их также можно рассматривать как полные таймеры общего назначения. Их 4 независимых канала могут использоваться для генерации ШИМ (режимы с выравниванием по краю или по центру) для 4-х колес соответственно.

Учитывая 16-битность таймеров, можно достичь 65536 уровней яркости, то есть более точно регулировать скважность для генерации ШИМ-сигнала. Изменяя скважность ШИМ-сигнала, можно контролировать уровень яркости светодиода или скорость вращения двигателя. Например, если ШИМ-сигнал

имеет скважность 50%, то светодиод будет светиться с половинной яркостью, а двигатель будет вращаться с половинной скоростью. [14]

Из рассмотренных таймеров следует, что предварительно лучше использовать расширенный таймер для своих нужд, в случае несовместимости можно перейти на другой таймер общего назначения без больших затрат, поэтому выберем таймер TIM8.

Джойстик

Используемый микроконтроллер будет получать данные и высчитывать необходимый момент и ток, следовательно, для проверки должно ли ускоряться или замедлятся колеса используется 2-х осевой джойстик (рисунок 8), где под угол наклона стика будет получено значение ускорения в некотором фиксированном диапазоне [15].



Рисунок 8 – Двухосевой джойстик

Будем использовать аналоговый джойстик, поэтому будет необходимо использование аналогово-цифрового преобразователя (АЦП) для чтения значений с джойстика и преобразования их в значения скважности. Затем использовать полученные значения для управления скважностью ШИМ-сигнала.

Использование 2-х осевого джойстика позволит организации управления не только вперед и назад, но также и для поворотов.

Оптопара

Для выбора оптопары есть некоторые ограничения, такие как как номинальное напряжение 12В от части батарей источника питания. Однако скорость оптопары является ключевым фактором для управления силовым ключом для “сглаженного” управления, минимальная ширина импульса ШИМ должна быть больше, чем скорость переключения оптопары [16].

Для возможных модификаций без просмотра возможных просадок на стороне оптопары можно использовать 6N137M (рисунок 9) [17].



Рисунок 9 – Оптопара 6N137M

Минимальная ширина импульса ШИМ должна быть больше, чем скорость переключения оптопары. Чтобы определить требуемую скорость оптопары, зная частоту ШИМ (F_{PWM}) и время нарастания (t_R) и спада (t_F) оптопары: $F_{PWM} = 2/n (t_R + t_F)$, где n - количество дискретных шагов, которые может обеспечить оптопара.

Датчик тока

Датчики ACS и ACS712 можно использовать для измерения тока в цепи, не влияя на производительность системы. Одной из вариаций датчиков тока на эффекте Холла является ACS758, где есть доступная модификация до 50А (ACS758LCB-050U-PFF-T). Более лучший вариант на 100А (рисунок 10) [18].

С помощью датчики тока на эффекте Холла можно без значительных влияний на цепь подключения измерять протекаемый ток благодаря поперечному магнитному полю [19].

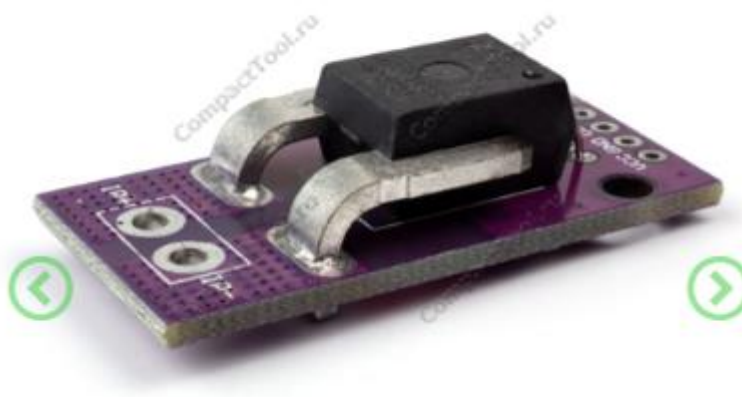


Рисунок 10 - ACS758LCB-100B-PFF-T

Подтягивающие и дополнительные резисторы

Что касается подтягивающих резисторов, то они могут быть необходимы в зависимости от типа оптопары и силового ключа. Для оптопары может потребоваться подтягивающий резистор на входе для установки нужного уровня логического сигнала. Для силового ключа может потребоваться подтягивающий резистор на затворе для установки нужного уровня напряжения затвора.

Для расчета сопротивления подтягивающего резистора на входе оптопары можно использовать формулу: $R = (V_{cc} - V_f) / I_f$, где V_{cc} - напряжение питания, V_f - прямое падение напряжения на светодиоде оптопары, I_f - прямой ток светодиода.

Для расчета сопротивления подтягивающего резистора на затворе силового ключа можно использовать формулу: $R = V_{cc} / I_g$, где V_{cc} - напряжение питания, I_g - ток затвора.

Силовой ключ

Для силового ключа, с помощью которого будет регулироваться напряжение на ОБ может подойти IPB025N10N3 с работой до 180А, с учетом

лимита тока 100А с большим запасом, данного силового ключа будет достаточно [20].

Общение с системой управления якорной обмоткой

Для общения с другими микроконтроллерами и передачи пакетов данных используется CAN-шина, работающая с помощью модуля SN65HVD230 (рисунок 11).

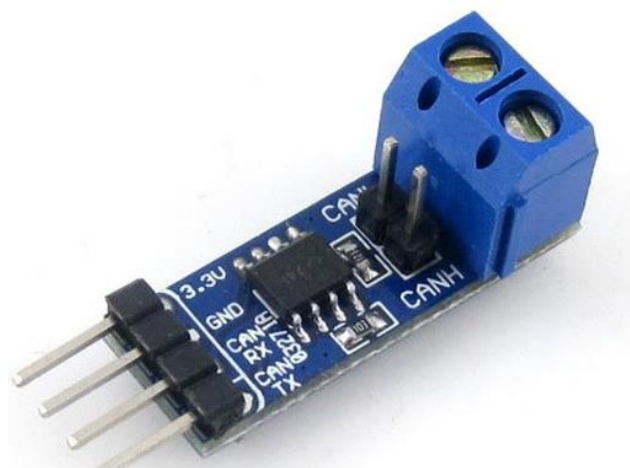


Рисунок 11 – модуль CAN SN65HVD230

Для подключения требуются терминирующие резисторы на двух крайних устройствах. Данный модуль уже имеет резисторы, поэтому для дополнительных устройств потребуется выпаять данные транзисторы со схемы подключения (рисунок 12) [21].

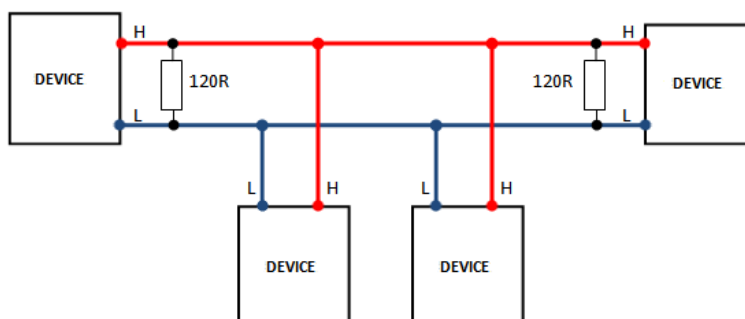


Рисунок 12 – Схема подключения устройств по CAN

Передача данных по шине CAN с использованием SN65HVD230 работает путем передачи и приема кадров данных между узлами в сети.

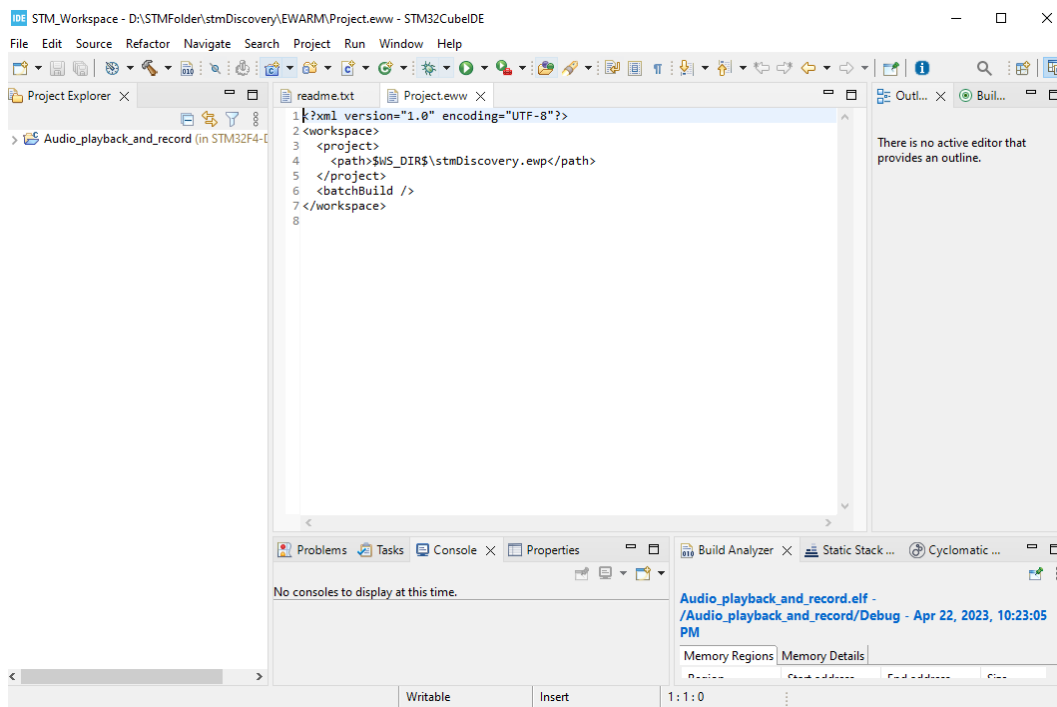


Рисунок 14 – Проект в CubeIDE

В случае с использованием GD32, то можно использовать следующие среды разработки:

- IAR Embedded Workbench – платный продукт для использования полного функционала, пробный период 14 дней [24];
- Keil MDK-ARM – платный продукт, пробный период 30 дней [25];
- Eclipse - бесплатная открытая среда разработки, которая может быть использована для разработки встраиваемых систем с GD32 микроконтроллерами. Однако для работы с GD32 может потребоваться установка дополнительных плагинов и инструментов [26].

Получение и обработка данных

Получение данных с отладочной платы на компьютер для анализа параметров мотор-колеса будет происходить на следующим шагам:

- отладочная плата подключена к компьютеру по имеющемуся порту USB;
- настроить проект для работы с отладочной платой и зашить используемую программу;

- программа должна иметь блок кода, который будет отправлять необходимые данные на компьютер;
- с помощью терминальной программы, такой как PuTTY настраивается виртуальный COM-порт. Терминальная программа используется для удобной отладки процесса передачи и получения данных, получение данных в читаемом виде и тестирование работы программы через отправку данных;
- с помощью приложения на Python считывать и обрабатывать файл журнала, который сохраняет PuTTY как получаемые с отладочной платы данные, но можно миновать использование PuTTY благодаря библиотеке PySerial, которая служит для прямого чтения данных из последовательного порта отладочной платы.

4 РЕАЛИЗАЦИЯ УПРАВЛЕНИЯ И СХЕМЫ ПОДКЛЮЧЕНИЯ

Для реализации программы управления и возможно, дополнительных элементов управления необходимо составить полную схему системы управления.

На момент начала проектирования общей системы управления была реализована система управления якорной обмоткой для проверки работоспособности спроектированного мотор-колеса, моей задачей состоит дополнение системы управлением через внешний сигнал, то есть джойстик, получение показателей с МКяо, с датчиков на ОВ, расчет показателей, отправка данных, управление ОВ.

4.1 Элементы системы управления

Сформируем блок-схему подключения элементов всей системы управления в функциональном представлении. Для начала покажем работающую схему системы управления якорной обмоткой (рисунок 15).

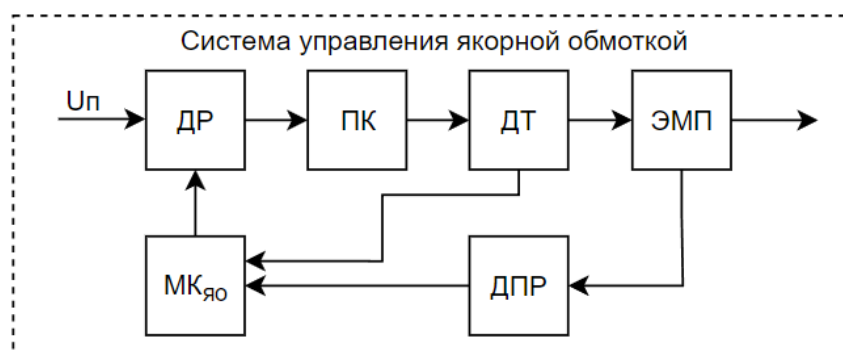


Рисунок 15 – Система управления якорной обмоткой

Спроектированная система имеет следующие элементы управления:

- ДР – драйвер, регулятор напряжения от источника и, следовательно, регулировка тока, подаваемая на двигатель;
- ПК – полупроводниковый коммутатор, реализующий 120-градусную дискретную коммутацию для работы двигателя;
- ДТ – блок датчиков тока, служит для реализации обратной связи и контроля текущего тока на якорной обмотке;
- ЭМП – электромагнитный преобразователь, то есть сам ВЭКВ;

- ДПР – датчик положения ротора, служит для правильной коммутации и расчета текущей скорости с некоторой погрешностью;
- МК – микроконтроллер реализующий логику управления системой управления.

Для тестового режима работы, указанной ранее схемы микроконтроллера заключалась в следующей общей схеме управления реализующая микроконтроллер якорной обмотки (рисунок 16).

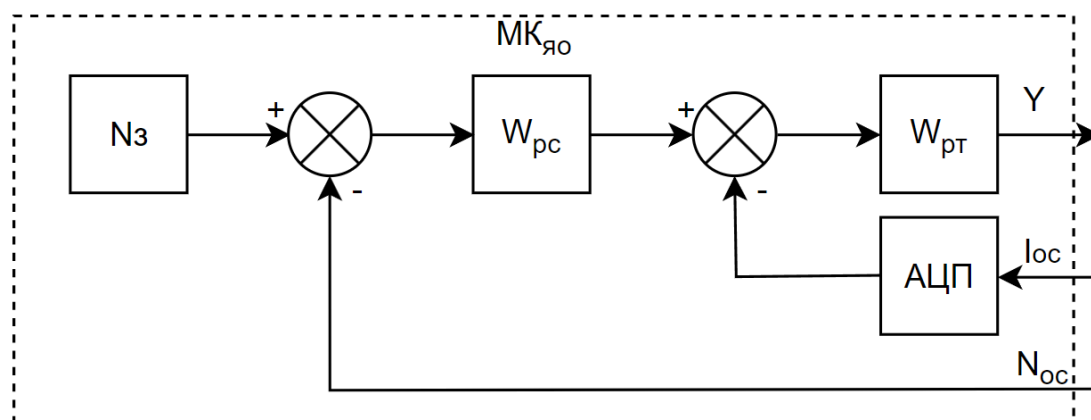


Рисунок 16 – Схема МК на якорной обмотке

Из данной схемы опишем задействованные элементы:

- N_z – Заданная скорость, которую необходимо достичь при тестировании на участках трассы;
- $W_{рс}$ – Передаточная функция расчета скорости;
- $W_{р\tau}$ – Передаточная функция расчета тока;
- Y – Значение скважности, с помощью которой регулируется напряжение и, следовательно ток, на якорной обмотке;
- $I_{ос}$ – Аналоговое значение показателей датчика тока для реализации обратной связи;
- АЦП - Аналого-цифровой преобразователь для получения значения тока обратной связи;
- $N_{ос}$ – Значение текущей скорости, полученной от ДПР.

При реализации полноценной системы управления мы должны избавиться от расчета показателей на основе заданной скорости, которую необходимо достичь. Для этого используется джойстик, степень положения

джойстика определяет значение ускорения, а именно скважность, на основе которого будет рассчитана показатель скорости, с которой будет двигаться машина.

Спроектируем схему полную систему управления, заметим, что не реализована схема управления ОВ, поэтому проектирование данной схемы также входит в наши обязанности.

Существует несколько реализаций схемы управления ОВ. Описанная ранее схема ОВ подразумевает в себе возможность смены направления магнитного потока, следовательно, смены полярности, это необходимо для лучшей реализации управления, например, уменьшение магнитного потока от постоянных магнитов для замедления двигателя.

Но при текущей дискретной 120-градусной коммутации невозможно реализовать возможность смены полярности на ОВ, потому что в таком случае пропадет возможность корректно считывать данные с датчика Холла в следствии того, что полярность магнитов будет одинакова и будет невозможно обработать сигналы для дальнейшего управления.

Для текущей реализации возможно спроектировать схему управления ОВ с одним возможным направлением магнитного потока, но при будущей реализации векторного управления на ЯО возможна реализация смены магнитного потока, поэтому спроектируем два варианта управления, спроектированные схемы требуют практической проверки, поэтому не могут являться полностью достоверными. Первый вариант спроектирован с возможностью смены полярности (рисунок 17).

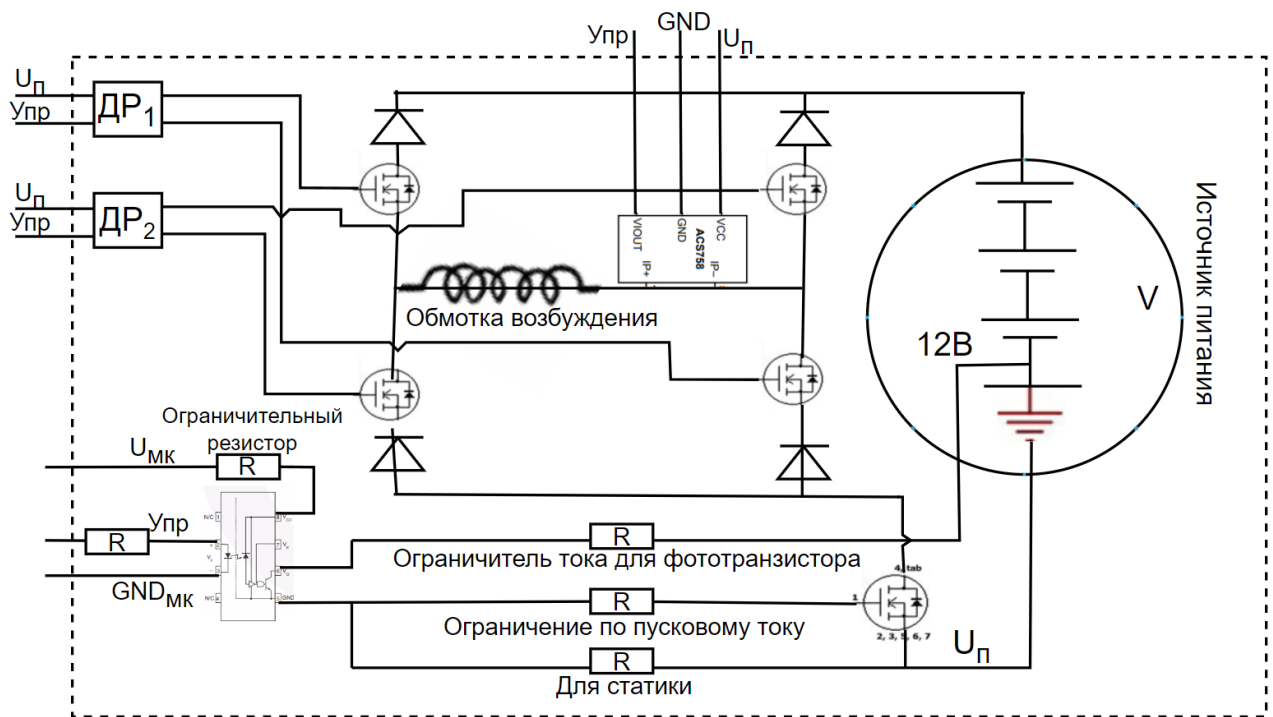


Рисунок 17 – Схема управления обмоткой возбуждения с возможностью смены полярности

Некоторые элементы показаны схематично и их реальная схема подключения может быть отлична от представленной, так как неопределенна модель управления ОВ. Данное количество датчиков тока обуславливается тем, что датчиком на фазе мы фиксируем текущий ток, а с помощью нижних и верхних датчиков контролируем возможный режим короткого замыкания для каждой из подключенной полярности, при этом сумма показаний со стоек одной пары датчиков также является потребляемым током, но для упрощения используется один на фазе, чтобы не дополнять программную логику по мониторингу тока.

Далее покажем схему подключения без возможности смены полярности (рисунок 18), которая скорее всего будет использована на первое время, пока не будет реализовано векторное управление.

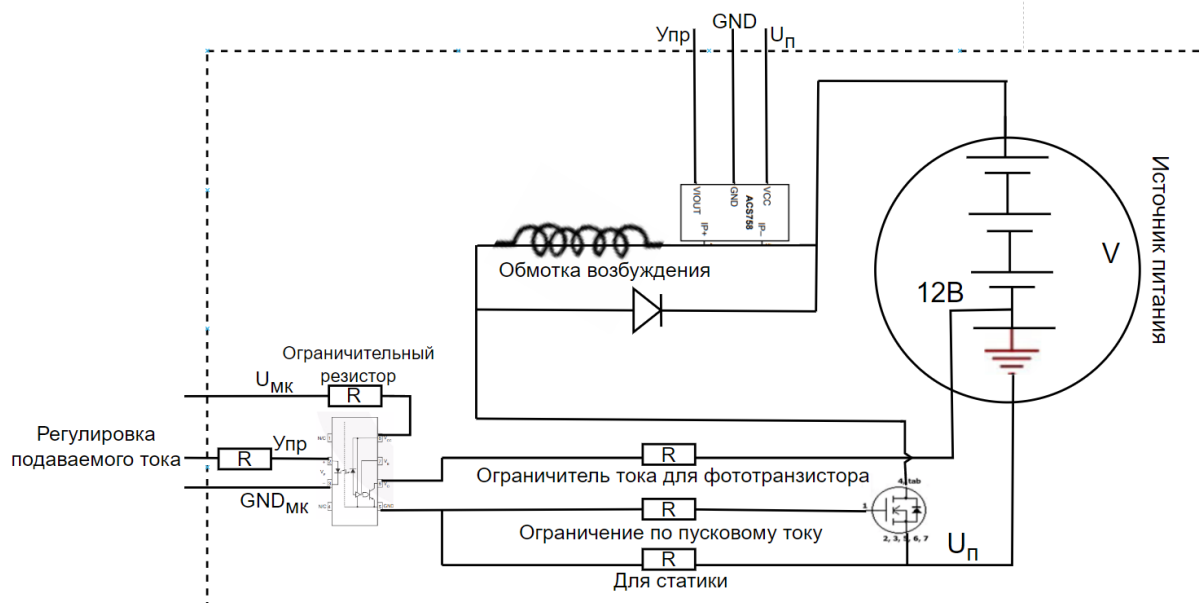


Рисунок 18 – Схема управления обмоткой возбуждения без возможности смены полярности

Управление с одной полярностью значительно упростило схему подключения, но при этом мы лишаемся дополнительного функционала по управлению, для составления общей схемы системы управления данный блок управления ОБ будет обозначен как блок “БУОВ” (блок управления обмоткой возбуждения). Покажем общую систему управления ВЭКВ (рисунок 19).

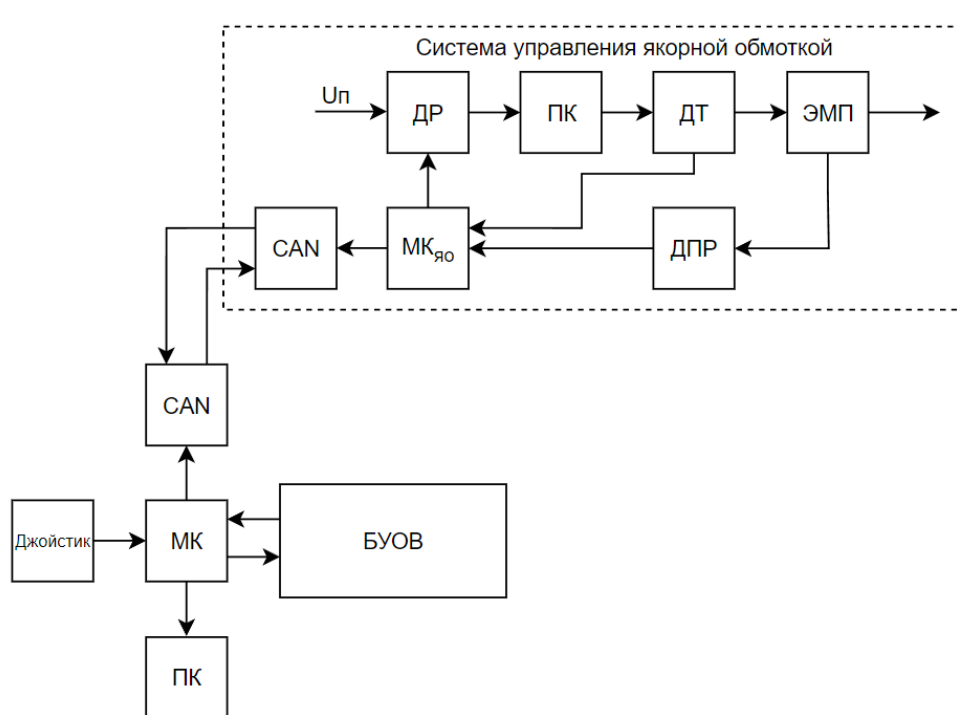


Рисунок 19 – Система управления ВЭКВ

Показана система управления для одного мотор-колеса, так как некоторые части, такие как система управления ЯО или структура ОВ реализованы, а дополнительные элементы организующие общую систему управления нет, то покажем подробную схему подключения элементов спроектированной части (рисунок 20).

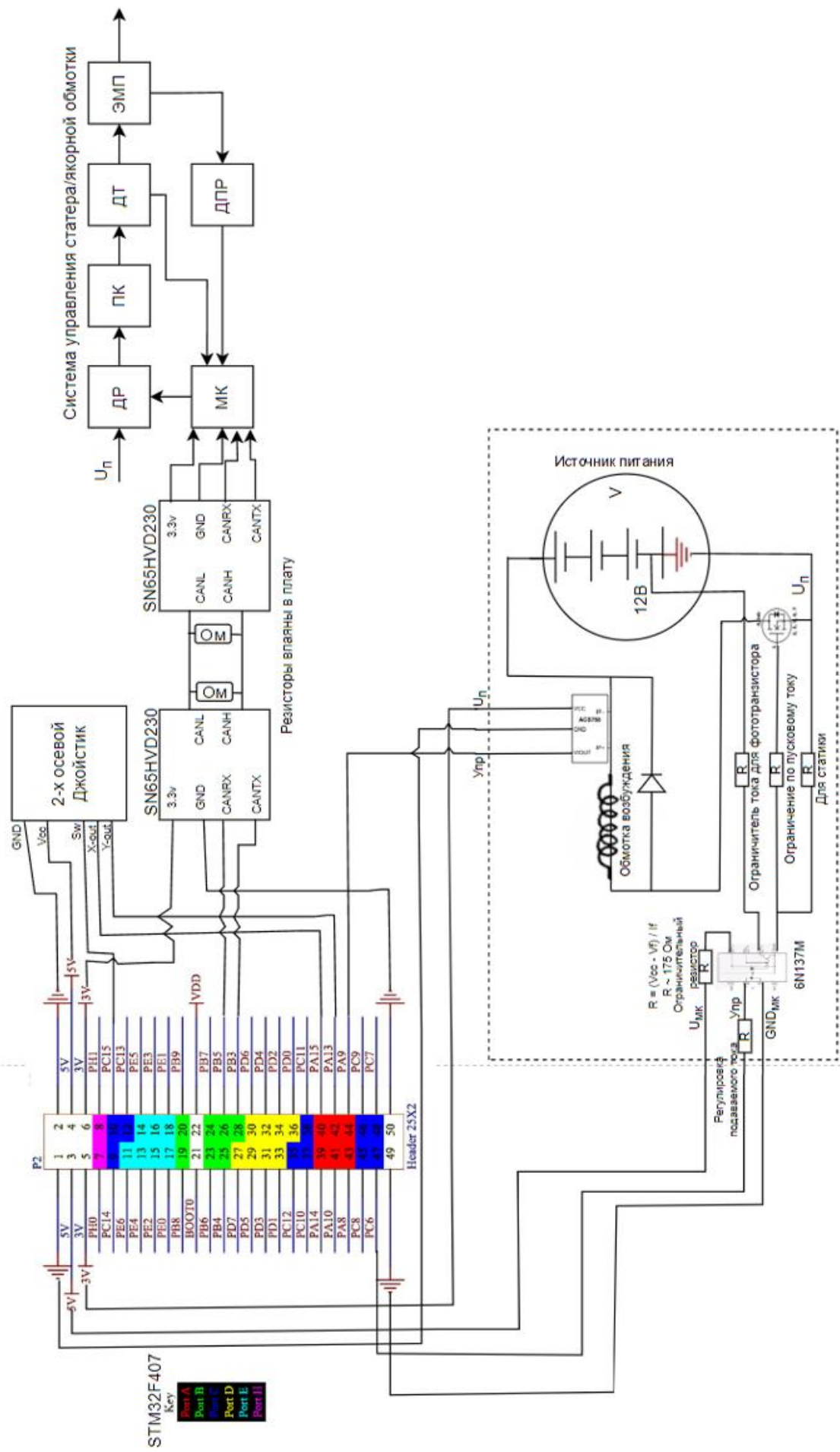


Рисунок 20 – Общая схема подключения системы управления

Представленная схема подключения с подробным указанием подключения элементов, которые были спроектированы в данной работе. Для демонстрации ОВ взята схема без возможности управления полярности.

На схеме не отображено подключение к персональному компьютеру, так как для этого не используются хедеры, а подключение происходит с помощью дополнительно порта micro-USB, который находится на самой отладочной плате.

Как можно увидеть на общей схеме подключения большинство портов платы МК не задействованы, поэтому возможно расширение для управления большим количеством колес, чтобы модифицировать управление с помощью джойстика для выполнения маневров поворота и т.д.

4.2 Алгоритм системы управления

Для данной работы реализована эмуляция управления с помощью “педали”, которую реализует джойстик. Изначально тестовые данные получались с помощью алгоритма, описанного ранее по заданным участкам и требуемой скоростью. Поэтому использованный алгоритм был адаптирован под текущие требования.

Блок получения задаваемых данных (рисунок 21) реализуется с помощью положения “педали”.

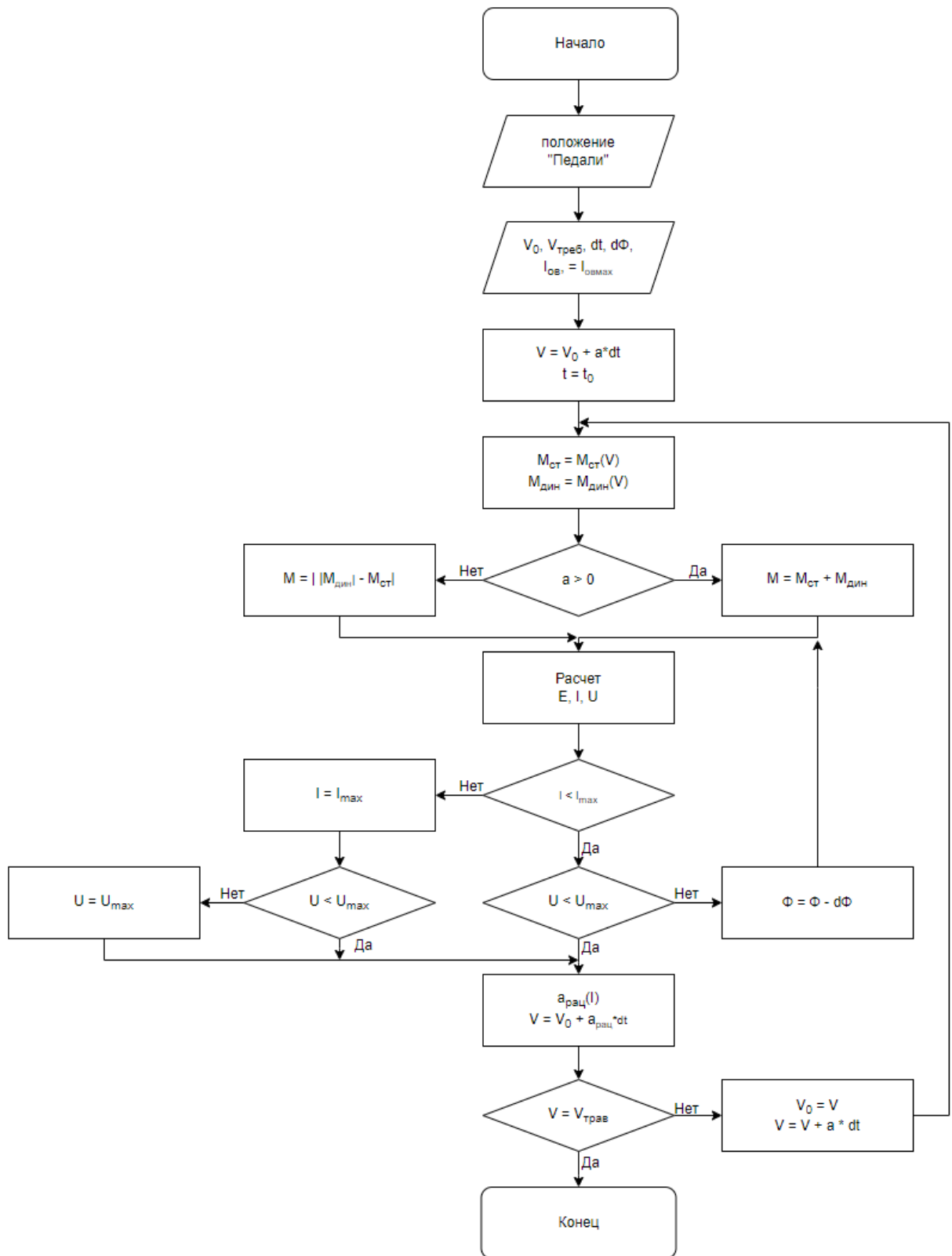


Рисунок 21 – Блок схема эмуляции управления “педалью”

Рассмотрев эмуляцию параметров болида на данных мотор-колесах были выявлены некоторые средние показатели на основе которых будет

реализована эмуляция, стоит заметить, что эмуляция проводится без учета наклона плоскости:

$$a_{max} = 17.4 \text{ м/с}^2 \text{ – максимальное ускорение;}$$

$$V_{max} = 36.2 \text{ м/с – максимальная скорость;}$$

$$t_{0-100} = 1.6 \text{ с – время разгона от 1 км/ч до 100 км/ч;}$$

$$V_{cp(0-100)} = 13.9 \text{ м/с – средняя скорость при разгоне (1-100) км/ч;}$$

$$S_{cp(0-100)} = 23 \text{ м – дистанция пройденная за разгон (1-100) км/ч.}$$

При полном нажатии на “педаль” будет подаваться указанные данные дистанции и средней скорости с дополнительными операциями по оптимизации достижения лимита скорости и режима, когда машина едет по инерции без ускорения.

Для этого необходимо настроить пины (рисунок 22) на используемом микроконтроллере для чтения положения “педали”, роль которой выполняет 2-х осевой джойстик, настроим максимально быстрое чтение данных пинов для лучшего отклика.

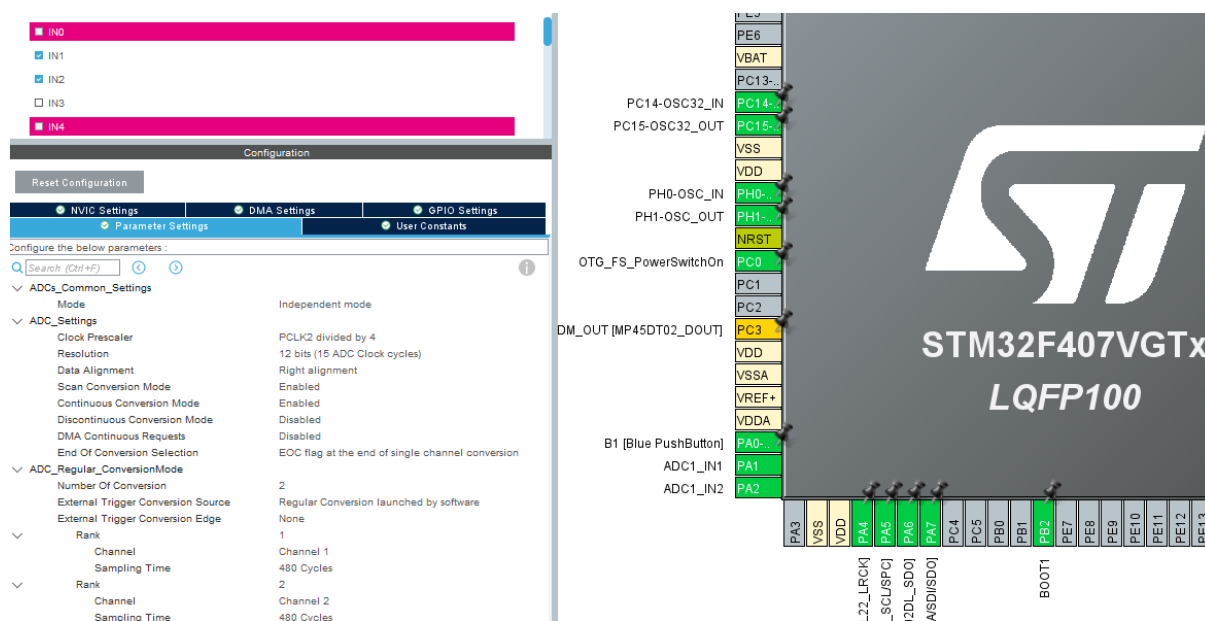


Рисунок 22 – Настройка пинов на чтение данных с 2-х осевого джойстика

Чтобы нам получать данные с нашего микроконтроллера настроена работа с виртуальным СОМ-портом, это необходимо для получения данных в не отладочном состоянии. У используемой отладочной платы есть

возможность реализовать данный обмен данных с помощью Mini-USB, используемый также для питания (рисунки 23-24).

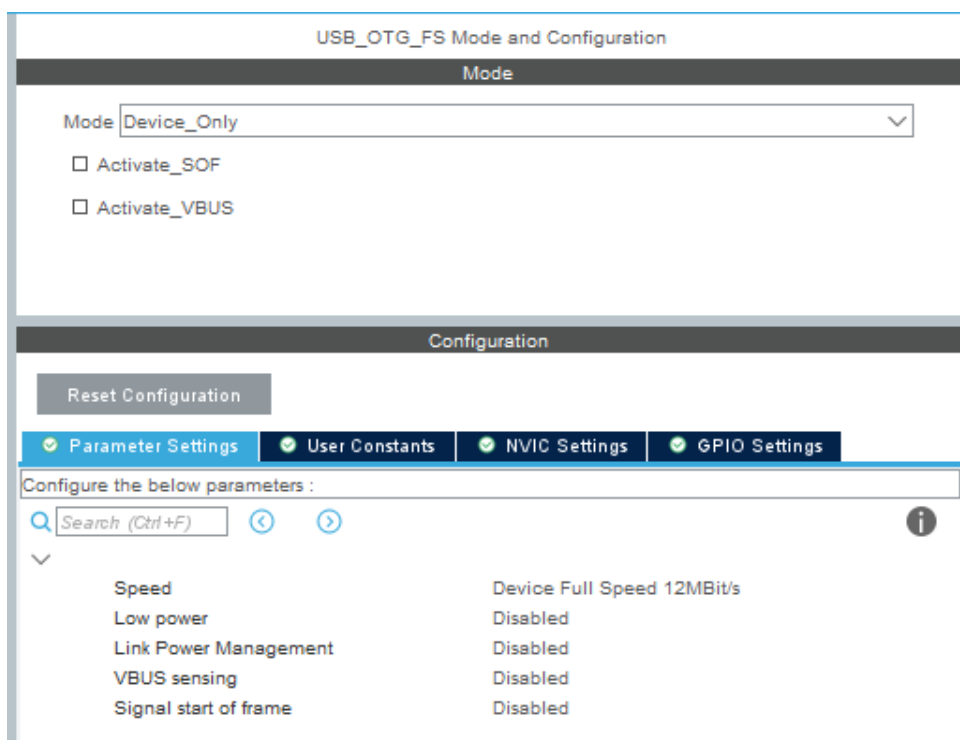


Рисунок 23 – Настройка типа подключения

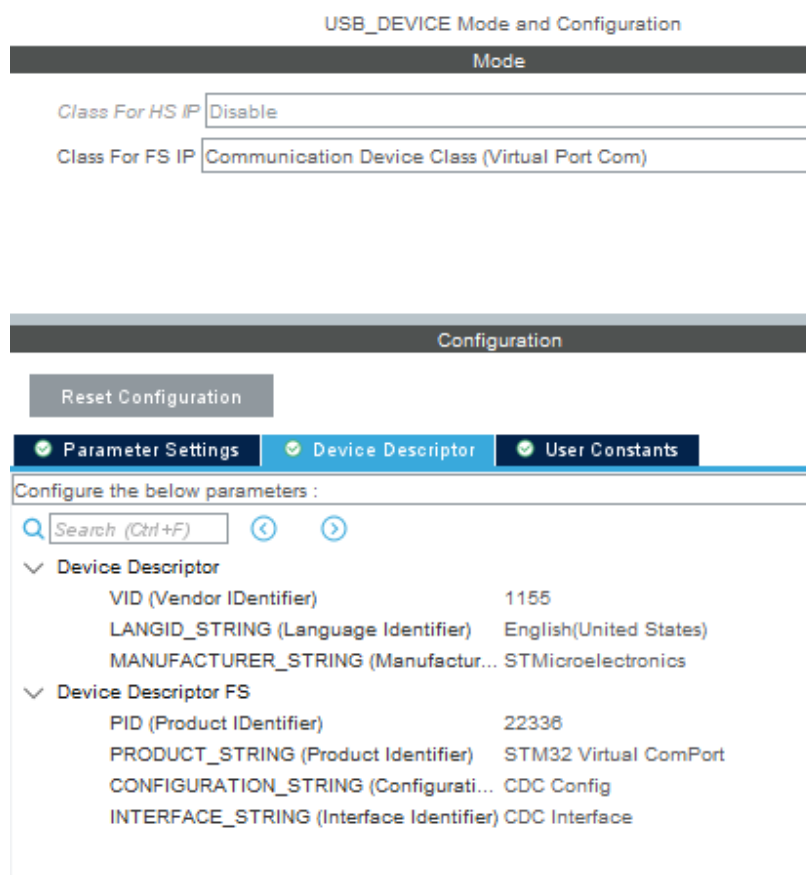
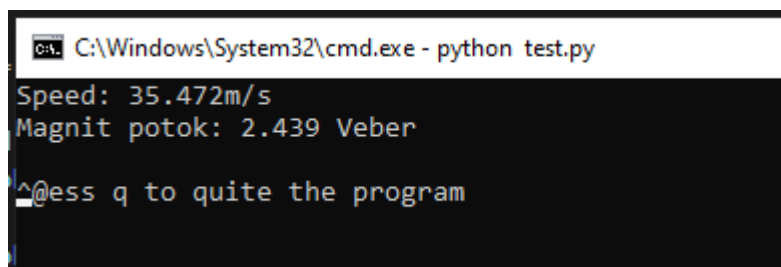


Рисунок 24 – Настройка USB подключения

После автоматической настройки библиотек и инициализируемых блоков кода реализован алгоритм работы с 2-х осевым джойстиком и передачей данных отображены в листинге А.1 приложения А, реализация кода на языке С.

4.3 Система мониторинга показателей

С помощью использования вспомогательных библиотек по визуализации и подключения к СОМ-порту было реализовано получение информации и отображение данных, реализация в листинге Б.1. Приложения Б. Для примера были получены с микроконтроллера данные о текущей скорости и суммарному магнитному потоку (рисунок 24).



```
C:\Windows\System32\cmd.exe - python test.py
Speed: 35.472m/s
Magnit potok: 2.439 Veber
^@ess q to quite the program
```

Рисунок 24 – Получение данных с СОМ-порта

Данные можно выводить на экран терминала из которого выполняется программа, данные приходят в виде строки с заданным нами разделителем, чтобы реализовать удобный парсинг строки и работать с данными как удобно. Полученные данные через визуальные библиотеки также можно отразить в качестве графиков или записать в .log файл для последующей обработки информации, если это потребуется.

ЗАКЛЮЧЕНИЕ

Данная система организует управление всеми блоками комбинированной машины, а именно регулирует микроконтроллер на якорной обмотке, который уже спроектирован, а также блок обмотки возбуждения, чтобы данные блоки работали автоматизировано уже от легкого ввода пользовательского интерфейса, в виде джойстика.

Система спроектирована для управления одного мотор-колеса, выбранный микроконтроллер и его аналог от GigaDevice имеет запас на подключение еще минимум 3-х мотор-колес, то есть расширение до 4-х, тем самым организовать полноценное управление всеми мотор-колесами. Для этого необходимо рассчитывать данные для каждого колеса отдельно с дополнительными условиями и положением джойстика для реализации поворота и т.д.

Для управления системой с 4-мя мотор-колеса скорее всего способна будет к подключению еще дополнительных элементов управления, такие как дополнительные датчики и т.д. В случае наличия дополнительных спроектированных блоков управления, то есть ABS и т.п., подключения не требует наличия дополнительных пинов на микроконтроллере, общение может происходить по CAN-шине.

Спроектированная система может осуществлять отладку текущих показателей болида в режиме реального времени. Система требует доработок и является базовой наработкой, по которой можно развивать полноценную систему управления с полной реализацией в соответствии с поставленными задачами.

БИБЛИОГРАФИЧЕСКОЙ СПИСОК

- 1 Лавру В.С. Источники энергии / В.С. Лавру. М. // Наука и техника, 1997. С. 67-145.
- 2 Применение электромеханической трансмиссии в грузовых автомобилях с целью улучшения экологических показателей / Р.Ю. Илимбетов, Е.В. Соломин, А.М. Астапенко, А.В. Баканов // URL: <https://cyberleninka.ru/article/n/primenenie-elektromehanicheskoy-transmissii-v-gruzovyh-avtomobilyah-s-tselyu-uluchsheniya-ekologicheskikh-pokazateley> (Дата обращения: 15.3.2023)
- 3 Синтез и анализ вентильных электродвигателей комбинированного возбуждения для электротрансмиссий наземных транспортных средств / И.А. Чуйдук
- 4 Arduino. Главная страница // URL: <https://www.arduino.cc/> (Дата обращения: 10.5.2023)
- 5 Raspberry Pi. Главная страница // URL: <https://www.raspberrypi.org/> (Дата обращения: 10.5.2023)
- 6 STM32 32-bit Arm Cortex MCUs // URL: <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html> (Дата обращения: 10.5.2023)
- 7 GigaDevice. Главная страница //URL: <https://www.gigadevice.com/> (Дата обращения: 10.5.2023)
- 8 All about RS485 – How RS485 Works and How to Implement RS485 into Industrial Control Systems? // URL: <https://www.seeedstudio.com/blog/2021/03/18/how-rs485-works-and-how-to-implement-rs485-into-industrial-control-systems/> (Дата обращения: 10.5.2023)
- 9 A Brief Introduction to Controller Area Network // URL: <https://copperhilltech.com/a-brief-introduction-to-controller-area-network/>
- 10 HubStub. STM32 ШИМ // URL: <https://hubstub.ru/stm32/81-stm32-shim.html> (Дата обращения: 10.5.2023)
- 11 Discovery kit with STM32F407VG MCU. User manual // URL: https://www.st.com/resource/en/user_manual/dm00039084-discovery-kit-with-stm32f407vg-mcu-stmicroelectronics.pdf (Дата обращения: 10.5.2023)
- 12 GigaDevice product selection guide 2021 // URL: <https://www.tme.eu/Document/a45554ff15ec17d572e7c9e01b3e71cc/GigaDevice-Selection-Guide-2021.pdf> (Дата обращения: 10.5.2023)
- 13 STMicroelectronics STM32 to GigaDevice GD32 // URL: <https://www.symmetryelectronics.com/blog/stmicroelectronics-stm32-to-gigadevice-gd32/> (Дата обращения: 10.5.2023)
- 14 Бражников А.М., Бражникова А.М., Забержинский Б.Э. О прецизионном преобразователе широтно-импульсного кода в напряжение // Научный результат. Информационные технологии. – Т.4, №4, 2019: // URL: [323516725.pdf](https://core.ac.uk/doi/pdf/10.26907/2541-7746.2019.4.323516725) (Дата обращения: 10.5.2023)

- 15 Обзор тактильного Джойстика // URL: <https://3d-diy.ru/wiki/arduino-datchiki/arduino-joystick/> (Дата обращения: 10.5.2023)
- 16 Optocoupler selection and usage for isolating a PWM // URL: <https://www.analogictips.com/selecting-optocoupler-isolate-pwm/> (Дата обращения: 10.5.2023)
- 17 3DiY. Принцип работы оптопары // URL: <https://3d-diy.ru/wiki/components/optopara/> (Дата обращения: 10.5.2023)
- 18 Документация на серию ACS758 // URL: <https://www.compel.ru/item-pdf/95747517b7ac27cb0dbe900971a71542/ps/allegro~acs758.pdf> (Дата обращения: 10.5.2023)
- 19 Современные промышленные датчики тока / Андрей Данилов // URL: <https://yadi.sk/i/TUskwQqGv2yKz> (Дата обращения: 10.5.2023)
- 20 Документация силового ключа IPB025N10N3 G // URL: https://www.infineon.com/dgdl/Infineon-IPB025N10N3G-DS-v02_03-en.pdf?fileId=db3a30431ce5fb52011d1ab1d9d51349 (Дата обращения: 10.5.2023)
- 21 CAN-шина и stm32 // URL: <https://istarik.ru/blog/stm32/159.html> (Дата обращения: 10.5.2023)
- 22 STM32CubeMX. Страница загрузки // URL: https://www.st.com/content/st_com/en/stm32cubemx.html#get_started_container (Дата обращения: 10.5.2023)
- 23 STM32CubeIDE. Страница загрузки // URL: https://www.st.com/content/st_com/en/stm32cubeide.html (Дата обращения: 10.5.2023)
- 24 IAR. Страница загрузки // URL: <https://www.iar.com/products/free-trials/> (Дата обращения: 10.5.2023)
- 25 Keil MDK. Страница загрузки // URL: <https://www.arm.com/products/development-tools/embedded-and-software/keil-mdk> (Дата обращения: 10.5.2023)
- 26 Eclipse IDE. Страница загрузки // URL: <https://www.eclipse.org/downloads/packages/release/2023-03/r> (Дата обращения: 10.5.2023)

ПРИЛОЖЕНИЯ

Приложение А

(Исходный код скриптов для эмуляции)

Листинг А.1 – Исходный пользовательский код для микроконтроллера

```
/* USER CODE BEGIN Includes */
#include "math.h"
/* USER CODE END Includes */

/* USER CODE BEGIN PV */
uint32_t inputXY[2];
uint8_t MSG[15] = {'\0'};
float acceleration;
float tempA;

//ПАРАМЕТРЫ АВТОМОБИЛЯ
float m = 400.0f; //Масса автомобиля
float D = 0.33f; //Диаметр колес
float jr = 10.0f; //Передаточное число редуктора.
float kls = 1.6f; //Площадь лобового сопр
float A = 1.6f; //Площадь лобового сопр
float kpdtrans = 0.9f; // Кпд трансмиссии
float fk = 0.021f; //коэф сопр кач от скорости
float alfa = 0.0f; //Наклон трассы
float J = 2.209f; //сум. момент инерции привода
float Jd = 0.0258f; // момент инерции ротора
float Nkolec = 2.0f; //количество колес

//ПАРАМЕТРЫ ДВИГАТЕЛЯ
float g = 9.81f; // ускорение свободного падения
float roair = 1.2f; //удельная плотность воздуха
float kb = 1.11f;
float deltj = 0.01f;
float pi = 3.14f;
float vz = 0.001f; // Воздушный зазор
float kdelta = 1.418f; //Коэф воздушного зазора
float Spol = 0.0007f; //Площадь поверхности полюса
float kmu = 1.485f; //Коэффициент насыщения
float hm = 0.004f; //Высота магнита
float Hc = 89000.0f; //Коэрцитивная сила
float Br = 1.1f; //Остаточная индукция
float ko = 0.98f; //Обмоточный коэф
float w1 = 8.0f; //Число пар полюсов
float p = 8.0f; //Число витков в фазе
float kvipr = 1.654f; //Коэф выпрямления
float ra75 = 0.00057f; //Актив сопр обмотки якоря
float Imax = 1000.0f; //Ограничение якорного тока
float Cm = 81.487f; //Постоянная момента
float mz = 1.89f; //Масса зубцов
float ma = 2.5f; //Масса спинки
float p50 = 3.5f; //Удельные потери в стали
float kdz = 5.2f; //Коэф магнит потерь на зубцах
float kda = 1.95f; //Коэф магнит потерь в спинке

//Обмотка возбуждения
float Unom = 72.0f;
```

```

float rov = 23.0f;
float Wov = 787.0f;
float V0, Time, Stotal, Iov, delIov, Fov, Vsr, Jsym, Jpr, Pa, Pov, f, Pzmag, Pzov,
Bvzmagm, Pamag, Paov, Pmag, Pdob, Psum, Rdelta, Rsum, Rmag;
float PotokPM, Jpo, Bvzmag, Upit, Iatreb, Fmag, adelt, delSi, Si;
float PotokOv, Bvz, Potoksum, nsr, E, n0, aaa, VT, nt, Mprivct, Mdin, Mtrans, Mdv,
Mpost, Msoprkach, Msoprpodem, Msoprvoz;
/* USER CODE END PV */

/* USER CODE BEGIN PFP */
extern uint8_t CDC_Transmit_FS(uint8_t* Buf, uint16_t Len);
extern void StartBody(float Sdist, float Vsr);
extern void FirstMethod();
extern void SecondMethod();
/* USER CODE END PFP */

/* USER CODE BEGIN 2 */
float Sdist = 0.1f;
float Vycsr = 0.05f;
float tempSr = 0.0f;
float tempDist = 0.0f;
float VycsrM = 240.0f;
float SdistM = 1657.0f;
Si = 0.0f;
/* USER CODE END 2 */

/* USER CODE BEGIN 3 */
HAL_ADC_Start_DMA(&hadc1, inputXY, 2); // start adc in dma mode for multichannel
if (inputXY[1] >= 2055) {
    tempA = inputXY[1] - 2055;
    acceleration = tempA / 2040;
    Sdist = 23.0f * acceleration + delSi;
    Vycsr = 50.0f * acceleration + (VT*3600.0f/1000.0f);
}
else {
    acceleration = 0.0f;

    if (Vycsr > 50.0f) {
        tempSr = Vycsr;
    } else {
        tempSr = 50.0f;
    }

    if (Sdist > 23.0f) {
        tempDist = Sdist;
    } else {
        tempDist = 23.0f;
    }

    Sdist = Sdist - (tempDist * 0.0075f);
    Vycsr = Vycsr - (tempSr * 0.022f);
}

if (Vycsr > VycsrM) {
    Vycsr = VycsrM;
}
if (Sdist > SdistM) {
    Sdist = SdistM;
}

```

```

if (Sdist <= 0.1f) {
    Sdist = 0.1f;
}

if (Vycsr <= 0.05f) {
    Vycsr = 0.05f;
}

if (VT <= 0.8f) {
    Si = 0.0f;
}

StartBody(Sdist, Vycsr);

sprintf(MSG, "%1.3f", VT);
CDC_Transmit_FS(MSG, sizeof(MSG));

HAL_Delay(10);
}
/* USER CODE END 3 */

/* USER CODE BEGIN 4 */
void StartBody(float Sdist, float Vycsr) {
    float delt = Sdist / (Vycsr * 1000.0f / 3600.0f);
    deltj = delt / 45.0f;
    aaa = (Sdist-V0*delt)*2/(delt*delt);
    VT = V0 + aaa * deltj; // конечная на участке скорость автомобиля в м/с
    if (VT < 0.0f) {
        VT = 0.0f;
        return;
    }
    Vsr = (VT + V0) / 2; // средняя скорость на элементарном участке
    Rdelta = vz * 2 * kdelta / (4 * pi * (float)0.000001 * Spol); // магнитное сопротивление воздушного зазора
    Rsum = Rdelta * kmu; // магнитное сопротивление всей магнитной цепи для ОВ
    Fmag = 2 * hm * Hc; // НС двух постоянных магнитов в цепи ПМ
    Rmag = 2 * hm * Hc / (Br * Spol); // магнитное сопротивление двух постоянных магнитов в цепи ПМ
    PotokPM = Fmag / (Rsum + Rmag); // магнитный поток по цепи постоянных магнитов
    n0 = V0 * 60 / (pi * D); // обороты двигателя в начале участка
    nt = VT * 60 / (pi * D); // требуемые обороты двигателя в конце участка
    adelt = aaa; // линейное ускорение на участке

    Msoprkach = fk * m * g * D * (float)cos(alfa) / 2; // момент сопротивления качения
    Msoprpodem = m * g * D * (float)sin(alfa) / 2; // момент сопротивления подъему в гору
    Msoprvozd = (float)0.5 * kls * roair * A * Vsr * Vsr * D / 2; // момент сопротивления воздушному потоку
    Mpost = Msoprkach + Msoprpodem + Msoprvozd; // суммарный статический момент
    Mprivct = Mpost / (jr * kpdtrans); // приведенный статический момент
    Jpo = J + m * D * D / 4; // Момент инерции вращающихся частей
    Jpr = Jpo / (jr * jr); // приведенный момент инерции автомобиля к валу двигателя
    Jsym = Jd + Jpr; // суммарный приведенный момент инерции автомобиля к валу двигателя
    Mdin = Jsym * 2 * adelt * jr / D; // суммарный приведенный к валу двигателя динамический момент
}

```



```

Mtrans = Mprivct + Mdin; // суммарный момент трансмиссии
Mdv = Mtrans / Nkolec;
Bvzmag = PotokPM / Spol; // индукция в зазоре от потока постоянных магнитов

Iov = Unom / rov; // ток обмотки возбуждения
delIov = Iov / 100.0f; // доля изменения тока возбуждения

FirstMethod();
}

void FirstMethod() {
    Fov = Iov * Wov; // НС обмотки возбуждения
    PotokOv = Fov / Rsum; // магнитный поток по цепи обмотки возбуждения
    Bvz = PotokOv / Spol; // индукция в зазоре от потока обмотки возбуждения
    Potoksum = PotokPM + PotokOv; // суммарный магнитный поток
    nsr = (nt + n0) / 2; // средние обороты двигателя на элементарном участке
    E = 4 * kb * ko * w1 * p * Potoksum * (nt + n0) * jr / (2 * 60); // ЭДС
    вращения в обмотке якоря
    Iatreb = Mdv / (Cm * Potoksum); // ток якоря, необходимый для создания момента
    Upit = E * (1.41f * kvipr) + Iatreb * ra75 * 2; // требуемое напряжение
    питания
    if (Upit < Unom) { // проверка напряжения на двигателе
        SecondMethod();
    } else {
        Iov = Iov - delIov; // изменение тока возбуждения
        FirstMethod();
    }
}

void SecondMethod() {
    if ((Iatreb > 0) && (Iatreb > Imax)) { // ограничение якорного тока
        Iatreb = Imax;
    }
    if ((Iatreb < 0) && (abs(Iatreb) > Imax)) {
        Iatreb = -Imax;
    }
    Mdv = Iatreb * (Cm * Potoksum);
    adelt = ((Mdv * Nkolec - Mprivct) * D) / (Jsym * 2 * jr); // Ускорение
    aaa = adelt;
    V0 = V0 + aaa * delTj;
    if (Vsr <= 0.0f) {
        Vsr = 0.01f;
    }
    if (V0 <= 0.0f) {
        V0 = 0.01f;
    }
    Pa = Iatreb * Iatreb * ra75 * 2; // электрические потери в якоре
    Pov = Iov * Iov * rov; // электрические потери в ОВ
    f = abs(p * (nt + n0) * jr / (2 * 60)); // частота перемагничивания в стали

    Pzmag = kdz * p50 * Bvzmag * Bvzmag * pow((f / 50), 1.3) * mz / 2; // потери
    в зубцах от потока ПМ
    Pzov = kdz * p50 * Bvz * Bvz * pow((f / 50), 1.3) * mz / 2; // потери в
    зубцах от потока ОВ
    Pamag = kda * p50 * Bvzmag * Bvzmag * pow((f / 50), 1.3) * ma / 2; // потери
    в спинке от потока ПМ
    Paov = kda * p50 * Bvz * Bvz * pow((f / 50), 1.3) * ma / 2; // потери в
    спинке от потока ОВ

```

Окончание приложения А

```
Pmag = Pzmag + Pzov + Pmag + Paov; // суммарные магнитные потери
Pdob = (float)0.005 * abs(Iatreb) * Upit; // добавочные потери
Psum = Pa + Pov + Pmag + Pdob; // суммарные потери

delSi = V0 * deltj + adelt * deltj * deltj / 2; // Приращение пути
на элементарном участке
delSi = Vsr * deltj;
Si = Si + delSi;
}
/* USER CODE END 4 */
```

Приложение Б

(Исходный код скриптов для обмена данными)

Листинг Б.1 – Исходный код скрипта для получения данных с микроконтроллера

```
import serial
import curses
import threading
import signal
import sys

global uart
uart = True

port = "COM4"
ser = serial.Serial(port, 115200, timeout=0)

screen = curses.initscr()

row = 0
col = 0
screen.addstr(row,col, "Speed: ")
screen.addstr(row + 3,col, "Press q to quite the program")
screen.refresh()

def my_Serial():
    value = float(0)
    average = float(0)
    sampling = 2
    sample = 0

    valuePokot = float(0)
    averagePotok = float(0)
    Potok = float(0)

    while uart:
        data = ser.readline()
        if len(data) > 0:
            try:
                data sensor = data.decode('utf8')
                formatted = data sensor.replace('\x00', '')
                formatted = formatted.split('|')

                if formatted[0] == ' ':
                    value = 0.00
                else:
                    value = float(formatted[0])
                if formatted[1] == ' ':
                    valuePokot = 0.00
                else:
                    valuePokot = float(formatted[1])

                average += value
                averagePotok += valuePokot
                sample += 1

            if sample == sampling:
```

```

        sensor = float(average / sampling)
        Potok = float(averagePotok / sampling)
        average = 0.000
        sample = 0

        screen.addstr(row, 7, '          m/s')
        screen.addstr(row, 7, '{:.3f}'.format(sensor))
        screen.addstr(row+1, 14, '          Veber')
        screen.addstr(row+1, 14, '{:.3f}'.format(Potok))
        screen.addstr(3, 0, '')
        screen.refresh()
        #print(data sensor)
    except:
        pass

def signal_handler(signum, frame):
    sys.exit()

signal.signal(signal.SIGINT, signal_handler)

t1 = threading.Thread(target=my Serial)
t1.daemon = True
t1.start()

while True:
    key = screen.getkey()
    if key == 'q':
        uart = False
        break

```