

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«АСТРАХАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Допускается к защите

«__» _____ 20__ г.

Зав. кафедрой ЭЭиА
А.Ю. Ключарев

Резцов Дмитрий Николаевич

**РАЗРАБОТКА МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ ОБНАРУЖЕНИЯ
РАЗЛИВОВ НЕФТЕПРОДУКТОВ С ПОМОЩЬЮ БПЛА**

Выпускная квалификационная работа – магистерская диссертация
по направлению **11.04.04 Электроника и наноэлектроника**
студента образовательной программы магистратуры
«Промышленная электроника и микропроцессорная техника»

Студент

подпись

Научный руководитель
к.ф.-м.н., доцент
кафедры ЭЭиА,
Декан факультета
ФМиИТ

Н.А. Выборнов

Астрахань 2020

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«АСТРАХАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ЗАДАНИЕ
на выполнение магистерской диссертации

студенту группы ПЭ-25 Резцову Дмитрию Николаевичу

1. Тема работы: «Разработка микропроцессорной системы обнаружения разливов нефтепродуктов с помощью БПЛА»
2. Цель работы: Разработка микропроцессорной системы обнаружения разливов нефтепродуктов с помощью БПЛА
3. Формулировка задания:
 - обзор аналогов систем обнаружения;
 - выбор датчиков освещенности;
 - разработка принципиальной схемы;
 - изготовление печатной платы;
 - программирование микрокомпьютера;
 - исследование работы устройства.

Проект ВКР должен предоставлен студентом в срок до «__» _____ 20__ г.

Научный руководитель ВКР «__» _____ 20__ г. _____ Н.А. Выборнов

Первый вариант ВКР предоставлен студентом в срок до «__» _____ 20__ г.

Научный руководитель ВКР «__» _____ 20__ г. _____ Н.А. Выборнов

Итоговый вариант ВКР предоставлен студентом в срок до «__» _____ 20__ г.

Научный руководитель ВКР «__» _____ 20__ г. _____ Н.А. Выборнов

Задание выдано студенту «__» _____ 20__ г. _____ Н.А. Выборнов

Задание принято к _____ «__» _____ 20__ г. _____ Д.Н. Резцов
исполнению студентом

АННОТАЦИЯ

Ключевые слова: группа роботов, разнородные роботы, система управления, техническое зрение, искусственный интеллект.

Данная магистерская диссертация содержит: 86 страницы, 60 рисунков, 40 литературных источников, 3 приложений.

Разработанный микропроцессорная система позволяет обнаруживать разливы нефтепродуктов и иные объекты. Разработана система обработки метаданных и привязки их к визуальным данным.

Методы исследования: анализ литературных источников, системный анализ, методы статистического анализа, методы математического моделирования, методы теории принятия решений, методы разработки надежного программного обеспечения, имитационное моделирование.

ANNOTATION

Keywords: group of robots, heterogeneous robots, control system, technical vision, artificial intelligence.

This master's thesis contains: 86 pages, 60 drawings, 40 literature, 3 applications.

The developed microprocessor system allows detecting oil spills and other objects. A system for processing metadata and their bindings to visual data has been developed.

Research methods: analysis of literary sources, system analysis, methods of statistical analysis, methods of mathematical modeling, methods of decision theory, methods for developing reliable software, simulation.

Содержание

ВВЕДЕНИЕ	2
1 АНАЛИТИЧЕСКАЯ ЧАСТЬ.....	4
1.1 Обзор существующих систем обнаружения разливов нефтепродуктов	4
1.2 Требования к разрабатываемой системе.....	9
2 КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ	11
2.1 Разработка структурной электрической схемы	11
2.2 Технологии передачи данных	12
2.2.1 UART	12
2.2.2 I2C	13
2.2.3 SPI	14
2.2.4 Сравнение технологий передачи	19
2.3 Выбор оборудования	20
2.3.1 Контроллер верхнего уровня.....	20
2.3.2 Контроллер нижнего уровня	25
2.3.3 Датчик барометр.....	29
2.3.4 Датчик компас	34
2.3.5 Датчик GPS/ГЛОНАСС	36
2.3.6 Спектральная и тепловизионная камера	41
2.4 Разработка принципиальной электрической схемы.....	44
2.5 Проектирование печатной платы системы управления станком	46
2.6 Изготовление печатной платы	Ошибка! Закладка не определена.
3 ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ	48
3.1 Математическое моделирование гибридного шагового двигателя	48
ЗАКЛЮЧЕНИЕ	70
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	71
ПРИЛОЖЕНИЕ А	72

ВВЕДЕНИЕ

Автоматизация технологического процесса — совокупность методов и средств, предназначенная для реализации системы или систем, позволяющих осуществлять управление самим технологическим процессом без непосредственного участия человека, либо оставления за человеком права принятия наиболее ответственных решений [1].

В последнее время появление новейших технологий, позволило создать высокоэффективные и инновационные установки, особого типа, предоставляющие возможность достичь должного уровня результата. Поэтому современному обществу, которое ведёт деятельность в различных сферах стало значительно проще достичь безукоризненного результата.

Автоматизация таких процессов как поиск объектов на большой области как производства, так и других участков обитания человека. Любое производство имеет объекты выбросов, которые не соответствуют нормам ПДК (предельно допустимая концентрация) и наносят вред окружающей среде.

В соответствии с Федеральным законом от 27 декабря 2019 года N 453-ФЗ в которых указаны объекты охраны окружающей среды (Ст. 4) нормативы предельных выбросов для промышленных объектов, а так же плата за негативное воздействие на окружающую среду (Ст. 16), мер ограничения или прекращения деятельности и т.д. Можно понимать какие экономические, временные и физические потери возникают при несвоевременном обнаружении техногенных происшествий.

Основным показателем экономики РФ по добыче сырья и его экспорте, показывает важность этого отраслевого рынка. В РФ расположены крупные источники добыча таких ресурсов страны. На данный момент в стране находятся более 130 нефтяных и конденсатных месторождений. Так же важно учитывать транспортно-логистические пути, которые наносят большой вред, окружающий и приводящий к потерям производства.

Существующие методы обнаружения разливов нефтепродуктов с помощью визуального осмотра группой людей с судна или вертолета, космической съемкой, БПЛА без спецоборудования и тд., имеют ряд ограничений:

- Большие финансовые расходы;
- Метод обнаружения;
- Время работы;

- Погодные условия.

Поэтому приоритетной задачей является изготовление микропроцессорных систем обнаружения нефтяных разливов с помощью БПЛА на основе нейронной сети и спектральной камеры.

Целью магистерской диссертации является разработка микропроцессорной системы обнаружения разливов нефтяных продуктов с помощью БПЛА.

Исходя из поставленной цели были сформированы следующие задачи:

- осуществить обзор существующих систем управления;
- выполнить патентный поиск микропроцессорных систем;
- разработать структурную электрическую схему;
- разработать принципиальную электрическую схему;
- спроектировать печатную плату управления;
- выполнить расчет надежности разработанной печатной платы;
- провести исследования работы разработанного устройства.

1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Обзор существующих систем обнаружения разливов нефтепродуктов

В решениях с возникающими вопросами, связанными с экологической безопасностью промышленных объектов нефтегазодобывающих предприятий, первостепенная задача заключается в своевременном обнаружении, оценке и предупреждении о загрязнении окружающей среды в результате несанкционированных сбросов и аварийных разливов нефти.

Реализация этих задач может быть решена регулярным аэрокосмическим зондированием и диагностическим обследованием наиболее подверженных опасности технических объектов, своевременным осуществлением предупредительных и ремонтно-восстановительных работ, экологического состояния природных объектов в зоне деятельности нефтегазодобывающих предприятий.

Аэрокосмический мониторинг включает комплекс дистанционных методов исследования, используемых в инженерно-экологических изысканиях, сочетающий многозональную и спектральнозональную аэрофотосъемку с материалами космических фото, сканерной, телевизионной, радиолокационной, инфракрасной и других видов съемок, осуществляемых с искусственных спутников Земли, орбитальных станций и пилотируемых космических кораблей.

В практике инженерно-экологических изысканий наиболее широко используют фото- и сканерные съемки.

На данный момент существует ряд дистанционных методик контроля, основанных на отличии оптических, тепловых и радиоактивных свойств воды, загрязненной нефтепродуктами, и чистой воды.

Дистанционные методики обнаружения нефтяных загрязнений можно разделить на пассивные и активные. В основах пассивных методиках лежат регистрация теплового (ИК и СВЧ) и естественного гамма излучения. При использовании активных методик исследования водной поверхности облучают источником излучения определенного спектрального состава с регистрацией излучения или флюоресценции. Наиболее доступной и дешевой методикой контроля состояния водной поверхности считается визуальной. Толщина пленки и объем разлившейся нефти могут быть установлены по внешнему виду самой пленки на водной поверхности.

Для обнаружения выбросов нефти в воду используют плавающие конструкции, а именно постоянно сканирующие устройства и буи. Оборудование такой типологии гораздо дешевле, чем дистанционное, поэтому его широко используют во многих районах мира.

Буи, расположенные в прибрежной и промышленной зоне оснащены сенсорами измерения температуры, давления и тд. Которые регистрируют данные под водой и в воздухе. На основе изменения полученных данных выводят аномальные скачки температуры. Но такой метод дает не достоверную информацию в своем анализе.

Метод обнаружения нефтяных разливов должен производить проверку при помощи дистанционного зондирования поверхности суши или воды. Но также важно учитывать экономические и временные расходы таких методов, при сохранении эффективности работы.

В статье Мельникова Г.С «МЕТОД И АППАРАТУРА ДИСТАНЦИОННОГО ОБНАРУЖЕНИЯ, РАСПОЗНАВАНИЯ И КОЛИЧЕСТВЕННОГО АНАЛИЗ РАЗЛИВОВ НЕФТИ НА МОРСКОЙ ПОВЕРХНОСТИ» описал систему обнаружения разливов нефти используя метод дистанционного зондирования морской поверхности ИК-аппаратурой.

Наряду с радиолокационным двух-спектральным методом обнаружения нефтяных разливов, метод одновременного наблюдения ВПМ на длинах волн 7,5...14 мкм и 3,2...5,6 мкм однозначно обеспечит контрастное выделение разлива нефти за счёт сглаживания флуктуаций волнения моря и за счёт локального изменения линейного градиента температуры в верхнем скин слое воды сопутствующем и обусловленном тепло-массообменном между средами «вода-атмосфера».

В начальной стадии растекание нефти на поверхности воды обусловлено действием удельного веса, которому противостоит сила инерции. После растекания нефти до критической толщины 8 мм, наиболее важным фактором, способствующим распространению нефти, становится поверхностное натяжение. Растекание происходит наиболее быстро, если поверхностное натяжение воздух/вода больше, чем сумма поверхностных натяжений воздух/вода и нефть/воздух. Теоретически сырая нефть может растекаться до образования мономолекулярного слоя, хотя в естественных условиях этого, очевидно, не происходит. Распространение нефтяной пленки тормозится тонким слоем воды.

Этот скин-слой обусловлен непрекращающимся тепло-массо обменом сред «вода-атмосфера» представляет собой плёнку с ламинарным движением жидкости в нём и линейным профилем температурного градиента.

Когда толщина нефтяного слика становится равной толщине этого водного слоя, главным фактором, препятствующим растеканию, становится вязкость, в связи, с чем скорость растекания снижается. Такое положение достигается примерно в течение часа. Как только толщина слика достигает значения меньше критического, факторы, влияющие на растекание, уже не будут зависеть от объема вылитой нефти. Теоретические оценки толщины слика после разлива 100 м³ сырой нефти в зависимости от времени, прошедшего от момента разлива, приведены в табл. 1.

Таблица 1.

Время, сек	10 ²	10 ³	10 ⁴	10 ⁵
Толщина, мм	2 ÷ 3	0,5 ÷ 0,7	0,1 ÷ 0,15	0,02 ÷ 0,03

Экспериментальные исследования из космоса показали, что свежеразлитая нефть образует на поверхности воды пленку толщиной до 2 мм. Затем в течение 2 ÷ 5 суток пленка постепенно расплывается, достигая толщины 10 ÷ 100 мкм, которая определяется сортом нефти и температурой воды. Далее при контакте с водной поверхностью и воздухом теряются летучие и растворимые компоненты нефти. Оставшиеся тяжелые компоненты могут образовывать водную эмульсию, содержащую до 50 ÷ 80 % воды. Под воздействием волнения непрерывная нефтяная пленка расслаивается на отдельные полосы, параллельные большим ветровым волнам.

В методе, представленном Мельниковы Г.С. рассматривается оценка разлива нефти только на тепло-массообмене в ИК излучении при помощи тепловизионного оборудования, что дает не полный отчет об объекте исследования. И процесс не автоматизирован, что говорит о неэффективности и трудоемкости, и затрат времени.

Аэрокосмическая съемка дает весьма эффективный результат, но есть ряд сложностей и ограничений использования такого метода зондирования поверхности.

Во-первых, это погодные условия, которые дают ряд ограничений в связи с чем невозможно провести анализ.

Во-вторых, весьма ощутимые погрешности полученных данных, так как оптический анализ не производит идентификацию объекта, что в большой практике показывает путаницу между разливом нефтепродуктов и цветением водорослей.

В-третьих, аэрокосмическая съемка имеет цикл повторной съемки от 3 дней и время постобработки данных, что влечет большими потерями со стороны компании.

Аэрокосмическая компания Канопус-В (Россия) спутник (рис. 1) имеет ряд ограничений: цикл повторной съемки, разрешения пикселя, постобработка и стоимость съемки.

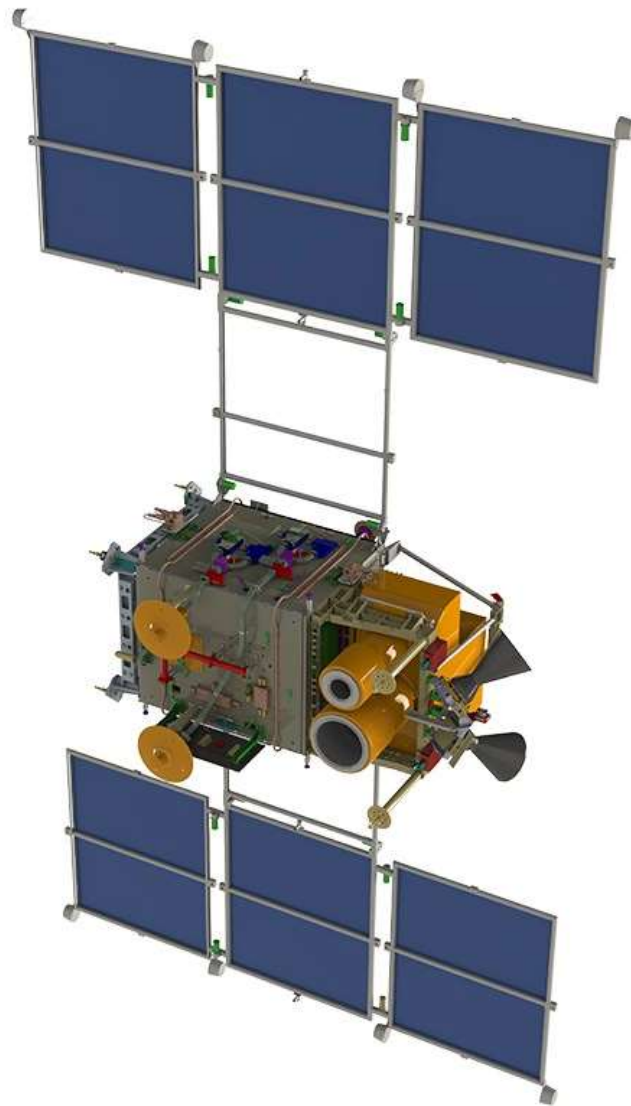


Рисунок 1.1 – Спутник Канопус-В

Таблица 1.2

Тип данных	оптические
Режим съемки	Моно
Спектральные каналы	панхроматический (0,54 – 0,86); синий (0,46÷0,52); зеленый (0,51, 0,6); красный (0,63÷0,69); ближний ИК (0,75 ÷ 0,84)
Пространственное разрешение в надире, м	2,5 м (панхроматический режим); 12 м (мультиспектральный режим)
Динамический диапазон, бит/пиксель	8
Ширина полосы съемки в надире, км	23 км (PAN) 20 км (MS)
Период повторной съемки	26 суток
Получение стереопары	нет

Аэрокосмическая компания SPOT 6&7 (Франция) спутник (рис. 2) характеристики которого представлено в таблице 3 и имеет ряд ограничений: цикл повторной съемки, разрешения пикселя, постобработка и стоимость съемки.

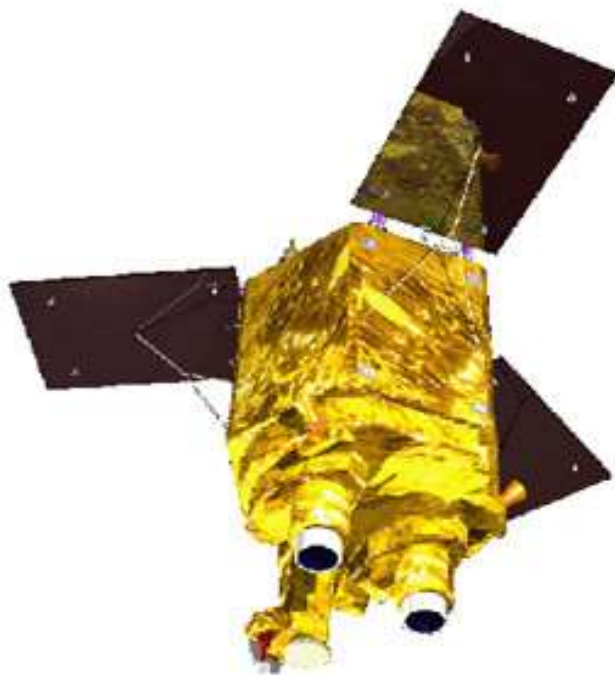


Рисунок 1.2 – Спутник SPOT 6&7

Таблица 1.2

Тип данных	оптические
Режим съемки	Моно и стерео съемка
Спектральные каналы	панхроматический, мультиспектральные (4 канала VNIR)
Пространственное разрешение в надире, м	1,5 (панхроматический); 6,0 (мультиспектральный)
Динамический диапазон, бит/пиксель	12
Ширина полосы съемки в надире, км	60
Период повторной съемки	ежедневно (в группировке)
Производительность съемки	более 6 млн (в группировке)
Получение стереопары	стереосъемка на одном витке, включая триплеты

1.2 Требования к разрабатываемой системе

К наиболее зарекомендовавшим себя методикам регистрации нефтяных разливов на водной поверхности на основе многочастотной аппаратуры радиолокационной съёмки с существенными дополнениями относятся методики многоканального дистанционного наблюдения в видимом и тепловом диапазонах спектра электромагнитных колебаний, а точнее спектрального анализа рис. 3. Который показывает радиохимический состав в разных спектральных каналах зондирования, что обуславливает качественный и эффективный метод анализа.

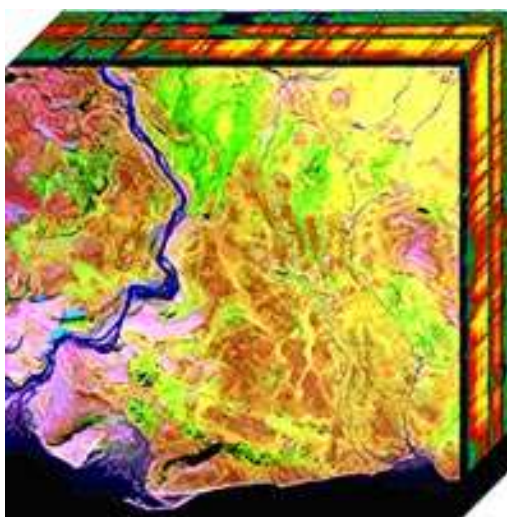


Рисунок 1.3 – Спектральный анализ поверхности

Одним из главных требований к разрабатываемой системе является полное автоматическое дистанционное обнаружение и идентификация разливов нефтепродуктов из полученных данных со спектральной и тепловизионной камеры, расположенные на БПЛА.

Для автоматического обнаружения важно понимать, что система должна ориентироваться себя в пространстве, а именно по GPS/ГЛОНАСС, знать высоту полета и углы наклона камер на стабилизирующих подвесе, и входить в согласование с полетным контроллером.

При разработке микропроцессорной системы обнаружения разлива нефтепродуктов с помощью БПЛА должно учитываться, что система будет работать в воздушном пространстве над водой и должна иметь влагонепроницаемый корпус и ударостойкость.

2 КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

2.1 Разработка структурной электрической схемы

Перед началом разработки самого устройства необходимо сначала построить структурную схему, с помощью которой мы сможем, верно, подобрать аппаратное обеспечение.

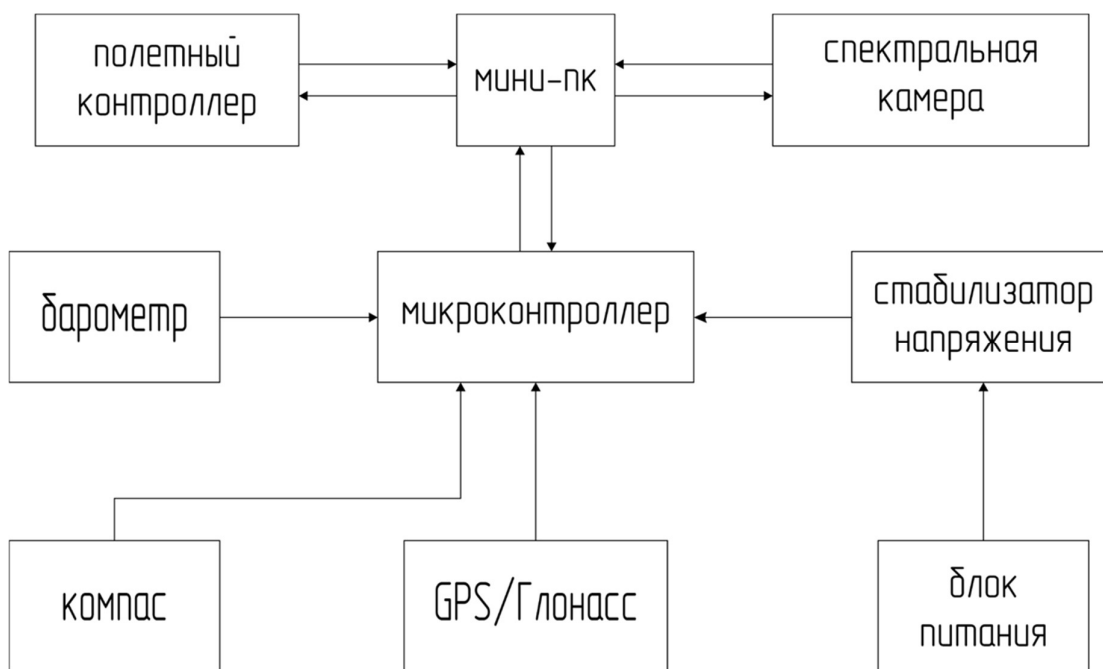


Рисунок 2.1 – Структурная схема микропроцессорной системы обнаружения разливов нефтепродуктов с помощью БПЛА

Разработанная структурная схема автоматического обнаружения (рисунок 2.1) имеет два уровня: верхний и нижний. Нижний уровень отвечает за прием данных с сенсоров, а именно получает текущие GPS координаты с точностью в ± 1 метр, высоту положения системы, и углы наклона камер. В качестве верхнего уровня выступает микрокомпьютер, который непосредственно обрабатывает полученные данные со спектральной и тепловизионной камеры, с нижнего уровня и полетного контроллера.

Верхний уровень получает поток кадров с камер частотой равной 1 секунде, программа обнаружения разливов нефтепродуктов, основанная на нейронных сетях, производит идентификацию объекта. Обработанные данные визуализируются объект, рассчитывается площадь и границы разлива. Данные, поступающие с нижнего уровня и полетного контроллера БПЛА, вычисляется достоверность и точность данных, формируются в пакет и высший уровень вычисляет границы разливов и формирует

полигон GPS координат, концентрацию загрязнения и идентификатор. Полученный пакет кодируется и отправляется по радиоканалу на наземную станцию, где происходит декодирование, прогнозирования последствий техногенного происшествия и отправка на сервер для дальнейшего реагирования и устранения последствий.

2.2 Технологии передачи данных

Разработанная система использует, как уже говорилось выше, два уровня, которые нужно между собой согласовать, поэтому встаёт вопрос о выборе технологии передачи данных между нижним и верхним уровнем. Так же нужно рассмотреть передачу сформированного пакета данных по радиоканалу.

2.2.1 UART

Универсальный асинхронный приёмопередатчик UART, та же известный как УАПП, является одним из самых распространенных интерфейсов в современной электронике. Протокол RS-232 является одним из самых известных из семейства UART. Родиной данного протокола, как ни странно, является США. Изначально данный протокол предназначался для отправки телеграфических сообщений. Механическое устройство являлось передатчиком. И для передачи данных использовалось всего пять бит. Но в современной электронике встречаются такие реализации UART, которые позволяют отправлять пять, шесть, семь, восемь, девять бит информации.

Все процедуры обмена данными по последовательному интерфейсу осуществляются посредством специальных микросхем, которые получили название универсальных асинхронных приемопередатчиков (от англ. UART — Universal Asynchronous Receiver/Transmitter). Все устройства, которые выполняют последовательный обмен данными, имеют тот или иной тип микросхемы UART. Тем не менее, аппаратная часть большинства таких приемопередатчиков базируется на топологии кристалла 8250 и его более продвинутой версии 8250A. Устройства серии 8250 включают такие популярные микросхемы асинхронных приемопередатчиков, как 16450, 16550, 16650 и т. д. Практически любую из этих микросхем можно обнаружить в персональных компьютерах и других коммуникационных устройствах [1].

Рассмотрим кратко основные порты UART. Основными линиями передачи и приема данных являются RXD и TXD. Порт RXD (Received Data) является приемником сообщений, а TXD (Transmitted Data) является передатчиком. Также

присутствуют другие порты, такие как DTS, RTS и пр. Но это уже зависит от версии UART. Приемник и передатчик работают независимо друг от друга, это означает, что UART является полнодуплексным интерфейсом.

Передача данных в UART осуществляется кадрами. Кадр — это последовательность бит фиксированной длины. Биты передаются в равные промежутки времени. Скорость UART и определяет данный интервал времени. Существует общепринятый ряд стандартных скоростей: 300- 600- 1200- 2400- 4800- 9600- 19 200- 38 400- 57 600- 115 200- 230 400- 460 800- 921 600 бод. Рассмотрим кадр поподробнее, он состоит из стартового бита, информационных битов, бита проверки четности, одного или двух стопбитов. В случае если количество информационных бит превышает максимальное количество возможных передаваемых битов, излишек битов будет передан следующей посылкой. Ключевым моментом в посылке является бит проверки четности. Ведь благодаря ему можно проверить целостность отправленной посылки.

Стоит пояснить, что такое бит четности. Битом четности в электронной вычислительной технике и сетях передачи данных называют контрольный бит, который служит для проверки чётности числа в двоичной системе счисления.

Пожалуй, самым сильным недостатком бита четности является его полная беспомощность при возникновении двух, четырех и т. д. ошибок в битах данных. Но что делать? Если канал передачи данных подвержен сильным помехам и практически, чуть ли не каждое сообщение приходит с множеством ошибок, а единственным интерфейсом для передачи данных является UART. Стандартная логика передачи данных UART уже будет бессильна.

2.2.2 I2C

I²C (IIC, англ. Inter-Integrated Circuit)[15] — последовательная асимметричная шина для связи между интегральными схемами внутри электронных приборов. Использует две двунаправленные линии связи (SDA и SCL), применяется для соединения низкоскоростных периферийных компонентов с процессорами и микроконтроллерами (например, на материнских платах, во встраиваемых системах, в мобильных телефонах).

Разработана фирмой Philips Semiconductors в начале 1980-х как простая 8-битная шина внутренней связи для создания управляющей электроники. Была рассчитана на частоту 100 кГц.

Стандартизована в 1992 году, в первой версии к стандартному режиму 100 кбит/с добавлен скоростной режим 400 кбит/с (Fast-mode, Fm); за счёт 10-битной адресации становится возможным подключение на одну шину более 1000 устройств, количество которых ограничивается максимально допустимой ёмкостью шины — 400 пФ.

В стандарте версии 2.0 (1998 год) представлены высокоскоростной режим работы 3,4 Мбит/с (High-speed mode, Hs) и требования пониженного энергопотребления. Незначительно доработан в версии 2.1 (2000 год).

В версии 3 (2007 год) добавлен режим 1 МГц (Fast-mode plus, Fm+) и механизм идентификации устройств (ID).

В версии 4 (2012 год) появился однонаправленный режим 5 МГц (Ultra Fast-mode, UFm) с использованием двухтактной логики без подтягивающих резисторов, добавлена таблица предустановленных идентификаторов.

В версии 5 (2012 год) исправлены ошибки.

В версии 6 (2014 год) пересчитаны графики, определяющие величину подтягивающих резисторов в зависимости от ёмкости шины и рабочего напряжения

IC использует две двунаправленные линии, подтянутые к напряжению питания и управляемые через открытый коллектор или открытый сток — последовательная линия данных (SDA, англ. Serial DAta) и последовательная линия тактирования (SCL, англ. Serial CLock). Стандартные напряжения +5 В или +3,3 В, однако допускаются и другие.

Классическая адресация включает 7-битное адресное пространство с 16 зарезервированными адресами. Это означает, что разработчикам доступно до 112 свободных адресов для подключения периферии на одну шину.

Основной режим работы — 100 кбит/с; 10 кбит/с в режиме работы с пониженной скоростью. Также немаловажно, что стандарт допускает приостановку тактирования для работы с медленными устройствами.

2.2.3 SPI

SPI (3-wire) - популярный интерфейс для последовательного обмена данными между микросхемами. Интерфейс SPI, наряду с I2C, относится к самым широко-используемым интерфейсам для соединения микросхем. Изначально он был придуман компанией Motorola, а в настоящее время используется в продукции многих производителей. Его наименование является аббревиатурой от 'Serial Peripheral Bus',

что отражает его предназначение - шина для подключения внешних устройств. Шина SPI организована по принципу 'ведущий-подчиненный'. В качестве ведущего шины обычно выступает микроконтроллер, но им также может быть программируемая логика, DSP-контроллер или специализированная ИС. Подключенные к ведущему шины внешние устройства образуют подчиненных шины. В их роли выступают различного рода микросхемы, в т.ч. запоминающие устройства (EEPROM, Flash-память, SRAM), часы реального времени (RTC), АЦП/ЦАП, цифровые потенциометры, специализированные контроллеры и др.

Главным составным блоком интерфейса SPI является обычный сдвиговый регистр, сигналы синхронизации и ввода/вывода битового потока которого и образуют интерфейсные сигналы. Таким образом, протокол SPI правильнее назвать не протоколом передачи данных, а протоколом обмена данными между двумя сдвиговыми регистрами, каждый из которых одновременно выполняет и функцию приемника, и функцию передатчика. Непременным условием передачи данных по шине SPI является генерация сигнала синхронизации шины. Этот сигнал имеет право генерировать только ведущий шины и от этого сигнала полностью зависит работа подчиненного шины.

Существует три типа подключения к шине SPI, в каждом из которых участвуют четыре сигнала. Самое простое подключение, в котором участвуют только две микросхемы, показано на рисунке 2.2. Здесь, ведущий шины передает данные по линии MOSI синхронно со сгенерированным им же сигналом SCLK, а подчиненный захватывает переданные биты данных по определенным фронтам принятого сигнала синхронизации. Одновременно с этим подчиненный отправляет свою посылку данных. Представленную схему можно упростить исключением линии MISO, если используемая подчиненная ИС не предусматривает ответную передачу данных или в ней нет потребности. Одностороннюю передачу данных можно встретить у таких микросхем как ЦАП, цифровые потенциометры, программируемые усилители и драйверы. Таким образом, рассматриваемый вариант подключения подчиненной ИС требует 3 или 4 линии связи. Чтобы подчиненная ИС принимала и передавала данные, помимо наличия сигнала синхронизации, необходимо также, чтобы линия SS была переведена в низкое состояние. В противном случае, подчиненная ИС будет неактивна. Когда используется только одна внешняя ИС, может возникнуть соблазн исключения и линии SS за счет жесткой установки низкого уровня на входе выбора подчиненной микросхемы. Такое решение крайне нежелательно и может привести к

сбоям или вообще невозможности передачи данных, т.к. вход выбора микросхемы служит для перевода ИС в её исходное состояние и иногда инициирует вывод первого бита данных.

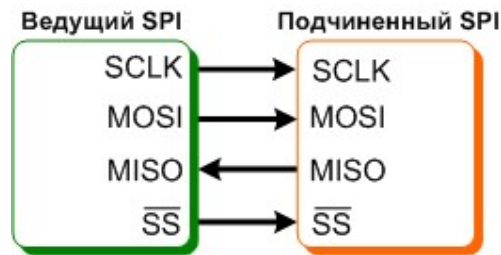


Рисунок 2.2 – Простейшее подключение к шине SPI

При необходимости подключения к шине SPI нескольких микросхем используется либо независимое (параллельное) подключение (рис. 2.3), либо каскадное (последовательное) (рис. 2.4). Независимое подключение более распространено, т.к. достигается при использовании любых SPI-совместимых микросхем. Здесь, все сигналы, кроме выбора микросхем, соединены параллельно, а ведущий шины, переводом того или иного сигнала \overline{SS} в низкое состояние, задает, с какой подчиненной ИС он будет обмениваться данными. Главным недостатком такого подключения является необходимость в дополнительных линиях для адресации подчиненных микросхем (общее число линий связи равно $3+n$, где n -количество подчиненных микросхем). Каскадное включение избавлено от этого недостатка, т.к. здесь из нескольких микросхем образуется один большой сдвиговый регистр. Для этого выход передачи данных одной ИС соединяется со входом приема данных другой, как показано на рисунке 2.4. Входы выбора микросхем здесь соединены параллельно и, таким образом, общее число линий связи сохранено равным 4. Однако использование каскадного подключения возможно только в том случае, если его поддержка указана в документации на используемые микросхемы. Чтобы выяснить это, важно знать, что такое подключение по-английски называется 'daisy-chaining'.

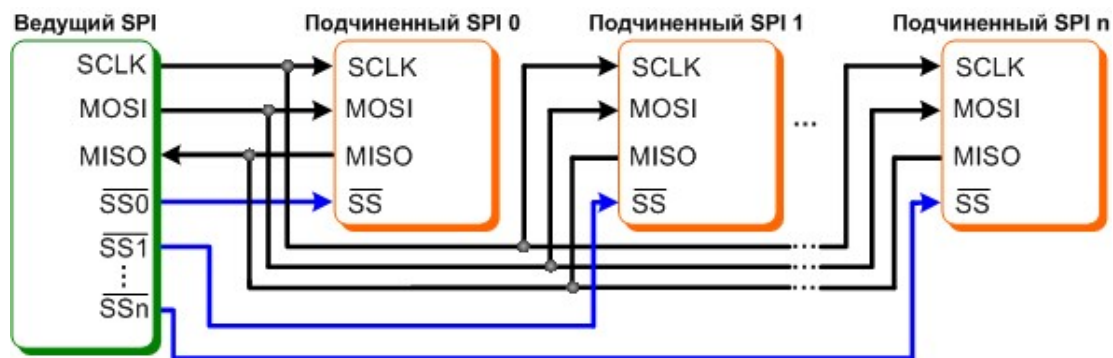


Рисунок 2.3 – Независимое подключение к шине SPI

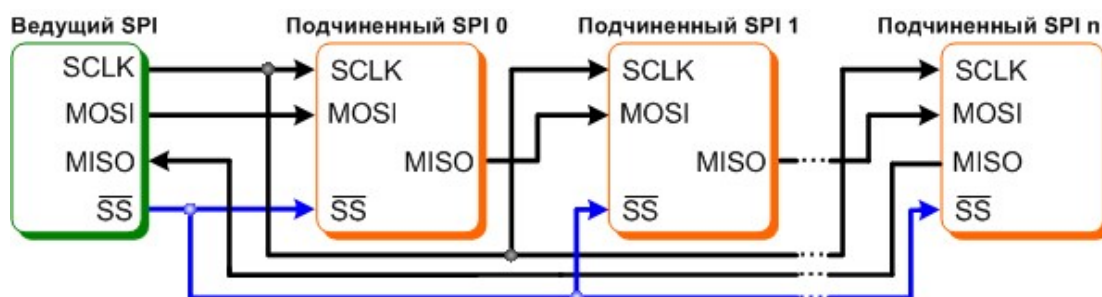


Рисунок 2.4 – Каскадное подключение к шине SPI

Протокол передачи по интерфейсу SPI предельно прост и, по сути, идентичен логике работы сдвигового регистра, которая заключается в выполнении операции сдвига и, соответственно, побитного ввода и вывода данных по определенным фронтам сигнала синхронизации. Установка данных при передаче и выборка при приеме всегда выполняются по противоположным фронтам синхронизации. Это необходимо для гарантирования выборки данных после надежного их установления. Если к этому учесть, что в качестве первого фронта в цикле передачи может выступать нарастающий или падающий фронт, то всего возможно четыре варианта логики работы интерфейса SPI. Эти варианты получили название режимов SPI и описываются двумя параметрами:

CPOL - исходный уровень сигнала синхронизации (если CPOL=0, то линия синхронизации до начала цикла передачи и после его окончания имеет низкий уровень (т.е. первый фронт нарастающий, а последний - падающий), иначе, если CPOL=1, - высокий (т.е. первый фронт падающий, а последний - нарастающий));

CPHA - фаза синхронизации; от этого параметра зависит, в какой последовательности выполняется установка и выборка данных (если CPHA=0, то по переднему фронту в цикле синхронизации будет выполняться выборка данных, а

затем, по заднему фронту, - установка данных; если же $CPHA=1$, то установка данных будет выполняться по переднему фронту в цикле синхронизации, а выборка - по заднему).

Ведущая и подчиненная микросхемы, работающие в различных режимах SPI, являются несовместимыми, поэтому, перед выбором подчиненных микросхем важно уточнить, какие режимы поддерживаются ведущим шиной. Аппаратные модули SPI, интегрированные в микроконтроллеры, в большинстве случаев поддерживают возможность выбора любого режима SPI и, поэтому, к ним возможно подключение любых подчиненных SPI-микросхем (относится только к независимому варианту подключения). Кроме того, протокол SPI в любом из режимов легко реализуется программно.

Как уже упоминалось, для стыковки микросхем не меньшей популярностью пользуется 2-проводная последовательная шина I2C. В таблице 3 можно ознакомиться с преимуществами, которая дает та или иная последовательная шина.

Производные и совместимые протоколы:

- MICROWIRE. Протокол MICROWIRE компании National Semiconductor полностью идентичен протоколу SPI в режиме 0 ($CPOL = 0$, $CPHA = 0$).
- 3-проводной интерфейс компании Maxim. Отличие этого интерфейса состоит в том, что вместо полнодуплексной передачи по двум однонаправленным линиям здесь выполняется полудуплексная передача по одной двунаправленной линии DQ.
- QSPI. Более высокоуровневый протокол, чем SPI, позволяющий автоматизировать передачу данных без участия ЦПУ.

Кроме того, интерфейс SPI является основой для построения ряда специализированных интерфейсов, в т.ч. отладочный интерфейс JTAG и интерфейсы карт Flash-памяти, в т.ч. SD и MMC.

Таблица 2.1

Преимущества шины SPI	Преимущества шины I2C
Предельная простота протокола передачи на физическом уровне обуславливает высокую надежность и быстродействие передачи. Предельное быстродействие шины SPI измеряется десятками мегагерц и, поэтому, она идеальна для потоковой передачи больших объемов данных и широко используется в высокоскоростных ЦАП/АЦП, драйверах светодиодных дисплеев и микросхемах памяти	Шина I2C остается двухпроводной, независимо от количества подключенной к ней микросхем.
Все линии шины SPI являются однонаправленными, что существенно упрощает решение задачи преобразования уровней и гальванической изоляции микросхем	Возможность мультимастерной работы, когда к шине подключено несколько ведущих микросхем.
Простота программной реализации протокола SPI.	Протокол I2C является более стандартизованным, поэтому, пользователь I2C-микросхем более защищен от проблем несовместимости выбранных компонентов.

2.2.4 Сравнение технологий передачи

Так как система оснащена не только передачей данных между верхним и нижним уровнем, а еще передачей данных между сенсорами и нижнем уровнем.

Сенсоры расположенные в микропроцессорной системе обнаружения разливов нефтепродуктов с помощью БПЛА имеют стандартные шины интерфейса I2C и SPI. В вышеуказанном разделе была приведена таблица 3 сравнения между двумя этими интерфейсами. Интерфейс SPI показывает простоту реализации, эффективность и надежность работы в отличии от I2C.

Передачей между микрокомпьютером и микроконтроллером эффективнее будет использовать интерфейс UART, хотя полное подключение всей системы к единой шине SPI будет лучше, но подключение к разным интерфейсам исключит факт возникновения пропуска пакетов данных.

2.3 Выбор оборудования

Микропроцессорная система обнаружения нефтепродуктов как на предприятии, так и в поле должна обладать большой степенью надежностью и отказоустойчивостью. Вследствие этого возникает вопрос в правильном выборе комплектующих, которые будут отвечать этим параметрам.

2.3.1 Контроллер верхнего уровня

Верхний уровень обрабатывает изображения полученные со спектральной и RGB камеры, при помощи нейронных сетей, т. е. искусственного интеллекта. Он должен обладать высокой производительностью и иметь графическое ядро GPU не ниже 256 ядер.

NVIDIA крупная компания производителей оборудования GPU ворвалась в эпоху искусственного интеллекта со встраиваемыми система, мини ПК и тд. Которые могут выполнять такие сложные вычисления и создавать технический прорыв в системах искусственного интеллекта широко применяемые в производстве, быту, логистике и тд.

Рассмотрим два мини-ПК от производителя NVIDIA, а именно JETSON TX2 SERIES (рис. 2.5) и JETSON AGX XAVIER SERIES (рис. 2.6).



Рисунок 2.5 – NVIDIA JETSON TX2



Рисунок 2.6 – NVIDIA JETSON AGX XAVIER

ПК серии JETSON TX2 построен на двухъядерном 64-битный процессоре NVIDIA Denver 1.5 и четырехъядерном процессоре ARM ® Cortex ® -A57 MPCore. GPU движок основан на 256-ядерном графическом процессоре NVIDIA Pascal™.

ПК серии JETSON AGX XAVIER SERIES построен на процессоре NVIDIA Carmel ARM®, который имеет 8 ядер, 8 МБ L2 и 4 МБ L3. GPU движок обладает высокими характеристиками 512-ядерный графический процессор NVIDIA Volta™ с 64 тензорными ядрами, в отличии от серии JETSON TX2.

В таблице 4 приведен сравнительный анализ двух мини-ПК компании NVIDIA серии JETSON TX2 и JETSON AGX XAVIER. В ней указаны основные параметры для систем, работающих с искусственным интеллектом. Искусственный интеллект данной системы основан на анализе получаемого видеопотока, поступающего с гиперспектральной и RGB камеры. Идентификации объекта на основе сет-листа с минимальными данными об объекте.

Таблица 2.2

Параметры	JETSON TX2	JETSON AGX XAVIER
ЦПУ	256-ядерный графический процессор NVIDIA Pascal™	512-ядерный графический процессор NVIDIA Volta™ с 64 тензорными ядрами
GPU	Двухъядерный 64-битный процессор NVIDIA Denver 1.5 и четырехъядерный процессор ARM® Cortex® -A57 MPCore	8-ядерный NVIDIA Carmel Arm® v8.2 64-разрядный процессор 8 МБ L2 + 4 МБ L3
Память	128 ГБ LPDDR4 8 ГБ (поддержка ECC) 51,2 ГБ / с	32 ГБ, 256-битный LPDDR4x 136,5 ГБ / с
Место хранения	32 ГБ eMMC 5.1	32 ГБ eMMC 5.1
Мощность	10 Вт / 20 Вт	10 Вт / 15 Вт / 30 Вт
CSI камера	До 6 камер (12 через виртуальные каналы) 12 дорожек MIPI CSI-2 D-PHY 1.2 (до 30 Гбит / с) C-PHY 1.1 (до 41 Гбит / с)	До 6 камер (36 через виртуальные каналы) 16 полос движения MIPI CSI-2 8 полос SLVS-EC D-PHY 1.2 (до 40 Гбит / с) C-PHY 1.1 (до 91 Гбит / с)
Кодирование видео	1x 4Kp60 3x 4Kp30 4x 1080p60 8x 1080p30 (H.265) 1x 4Kp60 3x 4Kp30 7x 1080p60 14x 1080p30 (H.264)	4x 4Kp60 8x 4Kp30 16x 1080p60 32x 1080p30 (H.265) 4x 4Kp60 8x 4Kp30 14x 1080p60 30x 1080p30 (H.264)

Декодирование видео	2x 4Kp60 4x 4Kp30 7x 1080p60 14x 1080p30 (H.265 и H.264)	2x 8Kp30 6x 4Kp60 12x 4Kp30 26x 1080p60 52x 1080p30 (H.265) 4x 4Kp60 8x 4Kp30 16x 1080p60 32x 1080p30 (H.264)
DL ускорение	Отсутствует	2x NVDLA двигателя
Vision ускорение	Отсутствует	7-Way VLIW Vision Processor

Сравнительный анализ показывает, что Мини-ПК NVIDIA серии JETSON AGX XAVIER обладает значительными преимуществами высокой производительности.

Интерфейс передачи данных у обоих мини-ПК идентичен и представлен ниже:

- UART;
- SPI;
- CAN;
- I2C;
- I2S;
- DMIC;
- GPIO;
- RJ45.

Последнее поколение высокопроизводительных компьютеров семейства Jetson AGX от NVIDIA, встраиваемых в Linux, Jetson AGX Xavier обеспечивает производительность класса рабочих станций на графических процессорах благодаря беспрецедентной производительности 32 TeraOPS (TOPS) и высокоскоростному вводу-выводу 750 Гбит / с в компактном корпусе. Пользователи могут настроить рабочие режимы на 10 Вт, 15 Вт и 30 Вт, как это необходимо для их приложений. Jetson AGX Xavier устанавливает новую планку для вычислительной плотности,

энергоэффективности и возможностей логического вывода ИИ, развертываемых на грани, позволяя интеллектуальным машинам следующего уровня с сквозными автономными возможностями.

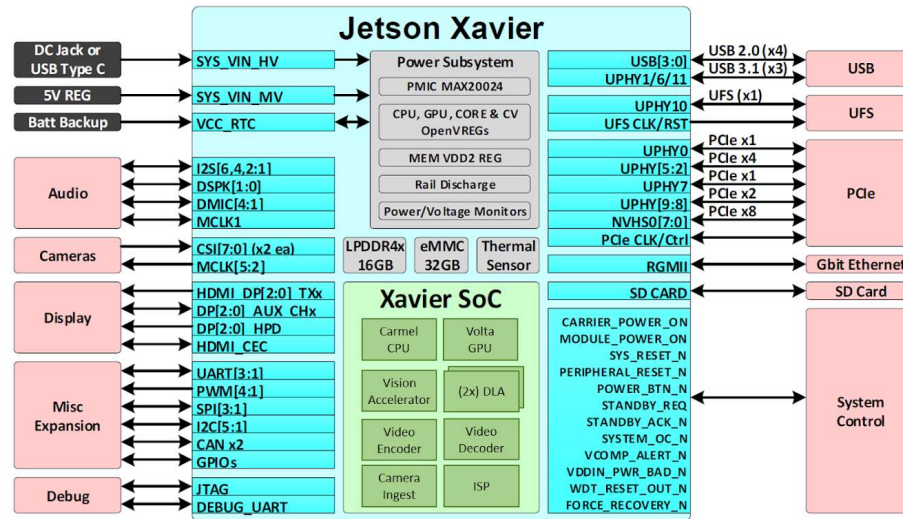


Рисунок 2.7 – Ввод/вывод Jetson AGX Xavier

Jetson AGX Xavier имеет более 750 Гбит / с высокоскоростного ввода-вывода, обеспечивая невероятную пропускную способность для потоковых датчиков и высокоскоростных периферийных устройств. Это одно из первых встроенных устройств с поддержкой PCIe Gen 4, обеспечивающее 16 линий на пяти контроллерах PCIe Gen 4, три из которых могут работать в режиме корневого порта или конечной точки. 16 линий MIPI CSI-2 могут быть подключены к четырем 4-полосным камерам, шести 2-полосным камерам, шести 1-полосным камерам или комбинации этих конфигураций до шести камер с 36 виртуальными каналами, позволяющими одновременно подключать, больше камер используя потоковую агрегацию. Другой высокоскоростной ввод / вывод включает три порта USB 3.1, SLVS-EC, UFS и RGMII для Gigabit Ethernet.

Тензорные ядра NVIDIA — это программируемые блоки с умножением и накоплением матрицы, которые работают параллельно с ядрами CUDA. Ядра Tensor (рисунок 2.8) реализуют новые инструкции HMMA (умножение и накопление матриц половинной точности) и IMMA (умножение и накопление матриц целых чисел) для ускорения вычислений с плотной линейной алгеброй, обработки сигналов и логического вывода с глубоким обучением.

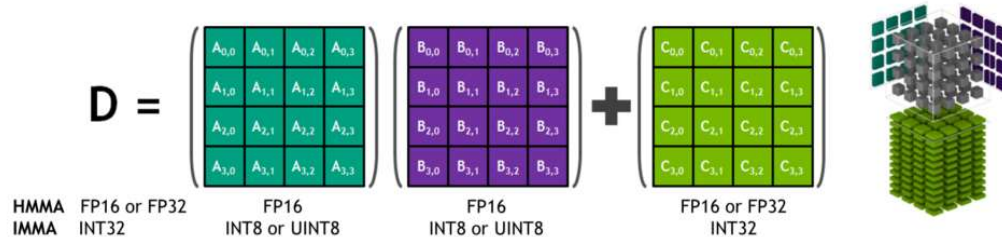


Рисунок 2.8 – Матрица Tensor Core HMMA / IMMA 4x4x4 умножается и накапливается

Матричные входы A и B умножения являются матрицами FP16 для команд HMMA, в то время как матрицы накопления C и D могут быть матрицами FP16 или FP32. Для IMMA матричный множитель A представляет собой матрицу INT8 или INT16 со знаком или без знака, B является матрицей INT8 со знаком или без знака, а матрицы-накопители C и D имеют INT32 со знаком. Следовательно, диапазон точности и вычислений достаточен, чтобы избежать переполнения и недостаточных условий во время внутреннего накопления.

Библиотеки NVIDIA, в том числе cuBLAS, cuDNN и TensorRT, были обновлены для внутреннего использования HMMA и IMMA, что позволяет программистам легко использовать преимущества прироста производительности, присущие тензорным ядрам. Пользователи также могут напрямую получать доступ к операциям Tensor Core на уровне деформации через новый API, представленный в пространстве имен wmma и заголовке mma.h, включенном в CUDA 10. Интерфейс уровня деформации отображает размеры 16×16 , 32×8 и 8×32 . Матрицы по всем 32 нитям на основе.

2.3.2 Контроллер нижнего уровня

Нижний уровень отвечает за обработку данных с сенсоров, а именно:

- Барометр, для вычисления высоты;
- Компаса, для определения углов наклона камер;
- GPS/ГЛОНАСС, для определения местоположения.

А также система должна формировать пакет данных, кодировать и отправлять на Мини-ПК (контроллер верхнего уровня). С помощью этих данных система будет определять свое местонахождение и границы разлива нефтепродуктов.

Микроконтроллер нижнего уровня будет выбран из двух семейств 8 – ми битовой AVR и 32 – ух битной ARM. В сравнении эти двух микроконтроллеров учитывается загруженность и частота опроса сенсоров, и передача полученных данных.

Микроконтроллером нижнего уровня из семейства AVR был выбран микропроцессор ATmega328p. Микроконтроллер ATmega328p является 8-ми разрядным CMOS микроконтроллером с низким энергопотреблением, основанным на усовершенствованной AVR RISC архитектуре.

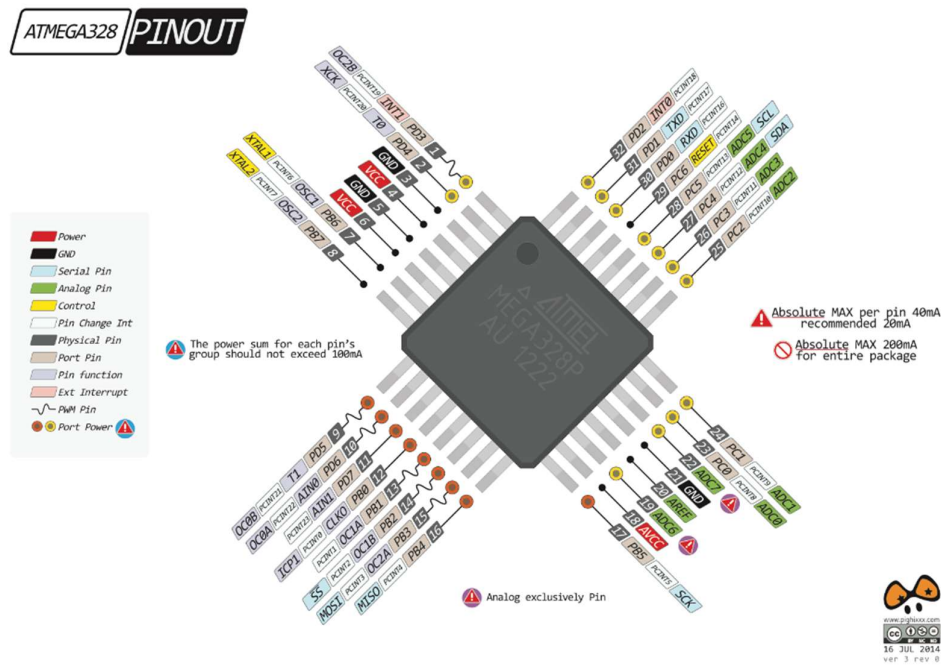


Рисунок 2.9 – Назначение контактов микроконтроллера ATmega328p

Огромное преимущество AVR-архитектуры — наличие 32 оперативных регистров, не совсем равноправных, но позволяющих в ряде случаев вообще не обращаться к оперативной памяти и не использовать стек (что в принципе невозможно в том же семействе x5\), более того, в младших моделях AVR стек вообще недоступен для программиста. Потому структура ассемблерных программ для AVR стала подозрительно напоминать программы на языке высокого уровня, где операторы работают не с ячейками памяти и регистрами, а с абстрактными переменными и константами.

Еще одна особенность AVR со схемотехнической точки зрения — все выводы в них могут пребывать в трех состояниях (вход — отключено — выход) и

электрически представляют собой КМОП-структуры (т. е. имеет место симметрия выходных сигналов и высокое сопротивление для входных). В общем случае это значительно удобнее двустабильных и TTL-совместимых портов и предполагает лучшую помехозащищенность (по крайней мере, от помех по шине "земли").

Таблица 1 – Характеристики ATmega328p

Производитель	Atmel
Серия	AVR® ATmega
Процессор	AVR
Размер ядра	8-Bit
Скорость	20MHz
Тип подключения	I ² C, SPI, UART/USART
Периферия	Brown-out Detect/Reset, POR, PWM, WDT
Число вводов/выводов	23
Размер программируемой памяти	32KB (32K x 8)
Тип программируемой памяти	FLASH
EEPROM Size	1K x 8
Размер памяти	2K x 8
Напряжение источника (Vcc/Vdd)	1.8 V ~ 5.5 V
Преобразователь данных	A/D 6x10b
Тип осцилятора	Internal
Рабочая температура	-40°C ~ 85°C
Корпус	TQFP

Микроконтроллером нижнего уровня из семейства ARM был выбран микропроцессор МК66FX1M0VMD18, а именно TEENSY 3.6. Микроконтроллер МК66FX1M0VMD18 является 32-ух разрядным CMOS микроконтроллером с низким энергопотреблением.

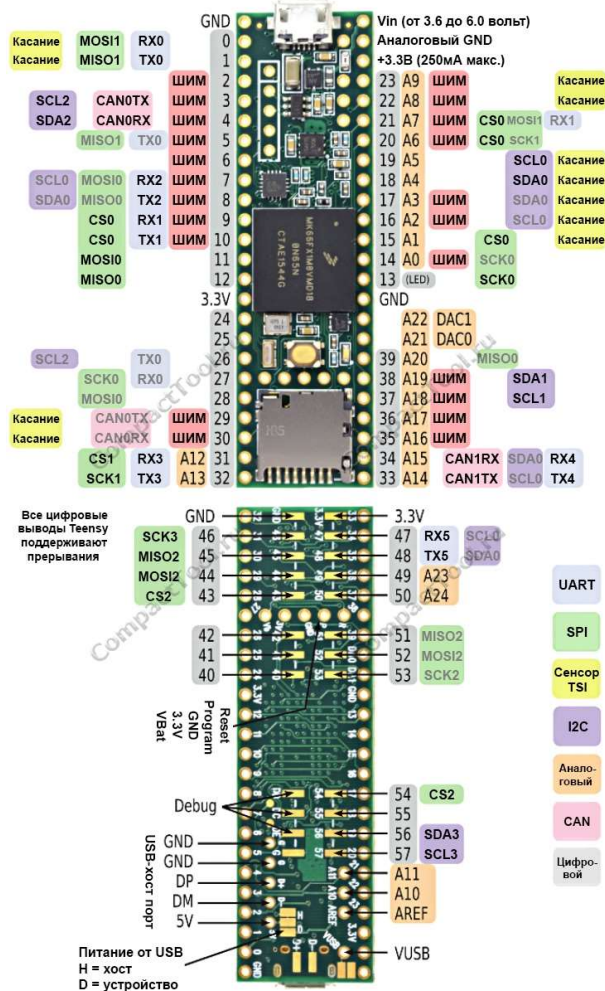


Рисунок 2.9. – Назначения пинов ввода/вывода Teensy 3.6

Версия 3.6 оснащена 32-битным процессором ARM Cortex-M4 с тактовой частотой 180 МГц и модулем с плавающей запятой. В ядре Cortex-M4 заложены DSP-инструкции (инструкции цифрового сигнального процессора), обеспечивающие достаточную мощность для реализации сложного алгоритма вычисления "Быстрого Преобразования Фурье" (спектральный анализ). Возможности DSP, входящего в состав Cortex-M4, позволяют параллельно выполнять четыре операции сложения/вычитания для 8-ми разрядных чисел или две операции сложения/вычитания с 16-ти разрядными операндами. Также реализовано умножение за один цикл, при этом для 16-ти разрядных чисел возможно параллельное исполнение двух операций.

Таблица 2 – Характеристики Teensy 3.6

Производитель	SparkFun Electronics
Процессор	ARM Cortex-M4
Размер ядра	32-Bit
Скорость	180 MHz
Тип подключения	I2C, SPI, UART/USART, I2S, CAN, Ethernet, USB
Периферия	Brown-out Detect/Reset, POR, PWM, WDT
Число вводов/выводов	57
Размер программируемой памяти	1024 KB
Тип программируемой памяти	FLASH
EEPROM Size	4096 KB
Напряжение источника (Vcc/Vdd)	3.6 V ~ 6 V
Преобразователь данных	A/D 6x10b
Тип осциллятора	Internal
Рабочая температура	-40°C ~ 85°C
Корпус	TQFP

Исходя из сравнительного анализа микроконтроллер основанный на 32 – ух битной архитектуре ARM Teensy 3.6 позволит проводить процесс считывания данных с сенсоров, их обработку и вычисления с помощью программных фильтров для отбраковки возникающих ошибок, быстрее и эффективнее.

2.3.3 Датчик барометр

Для данной работы был выбран самый оптимальный датчик в качестве цена качество - S2GO PRESSURE DPS310

DPS310 – это миниатюрный цифровой датчик атмосферного давления с высокой точностью и низким потреблением тока, способный измерять как давление, так и температуру. Элемент датчика давления основан на принципе емкостного сенсора, который гарантирует высокую точность при изменении температуры. Маленький размер делает DPS310 идеальным для мобильных устройств.

Внутренний сигнальный процессор преобразует выходные данные элементов датчика давления и температуры в 24-битные результаты. Каждая единица калибруется индивидуально, рассчитанные в ходе этого процесса калибровочные коэффициенты сохраняются в регистрах калибровки. Коэффициенты используются в программе для преобразования результатов измерений в высокоточные значения давления и температуры.

Результат FIFO (англ. first in, first out — «первым пришёл — первым ушёл») может хранить до 32 результатов измерений, что позволяет снизить частоту опроса хост-процессора. Измерения датчика и калибровочные коэффициенты доступны через последовательный интерфейс I2C или SPI. Состояние измерения указывается битами состояния или прерываниями на выводе SDO.

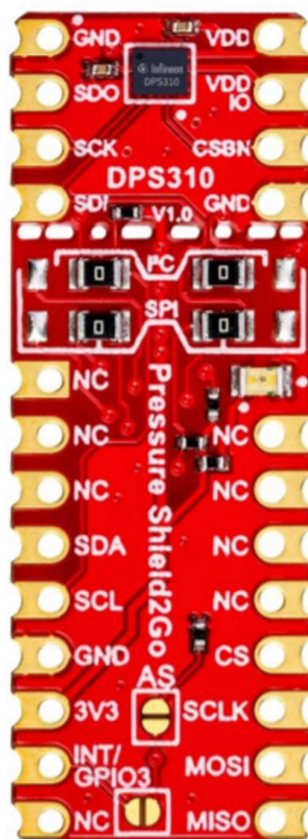


Рисунок 2.10 – Плата датчика давления и температуры DPS310

Основные технические характеристики датчика DPS310:

Рабочий диапазон: давление: 300–1200 гПа. Температура: -40 - 85 °С.

- Точность датчика давления: $\pm 0,005$ гПа (или $\pm 0,05$ м) (режим высокой точности);

- Относительная точность: $\pm 0,06$ гПа (или $\pm 0,5$ м);
- Абсолютная точность: ± 1 гПа (или ± 8 м);
- Точность температуры: $\pm 0,5$ °C;
- Чувствительность к давлению и температуре: 0,5 Па / К;
- Время измерения: типичное: 27,6 мс для стандартного режима (16х).
Минимум: 3,6 мс для режима низкой точности.
- Среднее потребление тока: 1,7 мкА для измерения давления, 1,5 мкА для измерения температуры при частоте дискретизации 1 Гц, в режиме ожидания: 0,5 мкА;
- Напряжение питания: VDDIO: 1,2–3,6 В, VDD: 1,7–3,6 В;
- Режимы работы: командный (ручной), фоновый (автоматический) и резервный;
- Калибровка: индивидуально откалибрована с коэффициентами для коррекции измерений;
- FIFO: сохраняет до 32 измерений давления или температуры;
- Интерфейс: I2C и SPI (оба с дополнительным прерыванием);
- Размеры корпуса: 8-контактный LGA, 2,0 мм x 2,5 мм x 1,0 мм;
- Соответствует экологическим требованиям (RoHS).

Доступ к DPS310 в качестве подчиненного устройства возможен через 3-проводной SPI-режим, 4-проводной SPI-интерфейс или последовательный интерфейс I2C.

Интерфейс I2C:

- Интерфейс датчика по умолчанию.
- Адрес датчика 0x77 (по умолчанию) или 0x76 (если вывод SDO опущен на GND).

Интерфейс SPI:

- Датчик переключится в конфигурацию SPI, если обнаружит активный низкий уровень на выводе CSB. SPI 4-wire является интерфейсом SPI по умолчанию.
- Чтобы включить трехпроводную конфигурацию SPI, бит должен быть установлен в регистре прерываний и конфигурации FIFO (CFG_REG) после запуска.

DPS310 может генерировать прерывание, когда доступен новый результат измерения и / или когда FIFO заполнен. Датчик использует вывод SDO для сигнала прерывания, и поэтому прерывание не поддерживается, если интерфейс является 4-проводным SPI. Прерывание включается и настраивается в регистре конфигурации прерываний и FIFO (CFG_REG). В конфигурации I2C вывод SDO служит как прерыванием, так и младшим значащим битом в адресе устройства. Если на вывод SDO подается низкий уровень, полярность прерывания должна быть установлена на активный высокий уровень, и наоборот. Состояние прерывания можно прочитать из регистра состояния прерывания (INT_STS).

После начала измерений самые последние необработанные данные о давлении и температуре будут доступны в соответствующих регистрах результатов. Измерение температуры можно пропустить. Измерения температуры можно отключить, если требуется быстрое измерение давления, но это сделает невозможной точную компенсацию температурного дрейфа. Все данные измерений могут быть прочитаны в одной команде с использованием считывания с автоинкрементом. Когда FIFO отключен, чтение регистра результатов не повлияет на значение регистра, оно будет обновлено только после завершения нового измерения. Когда FIFO включен, регистр результата давления будет обновляться до следующего значения в FIFO после каждого чтения. Когда все значения FIFO будут считаны, регистр результата будет установлен в 0x800000.

DPS310 FIFO может хранить последние 32 измерения давления или температуры. Это снижает общее энергопотребление системы, поскольку хост-процессору не нужно непрерывно опрашивать данные с датчика, но он может переходить в режим ожидания на более длительные периоды времени. FIFO может хранить любую комбинацию результатов измерения давления и температуры в соответствии с настройками скорости измерения в фоновом режиме. Например, скорость давления может быть установлена в 4 раза выше, чем скорость температуры, и, таким образом, только каждый результат будет температурным результатом. Тип измерения можно увидеть в данных результатов. Датчик установит младший значащий бит на:

- «1», если результатом является измерение давления.
- «0», если это измерение температуры.

Датчик использует 24 бита для хранения результата измерения. Поскольку это больше битов, чем необходимо для покрытия полного динамического диапазона

датчика давления, использование младшего значащего бита для маркировки типа измерения не повлияет на точность результата. FIFO можно включить в регистре конфигурации прерываний и FIFO. Данные из FIFO считываются из регистров данных давления (PRS_Vn) независимо от того, является ли следующий результат в FIFO измерением температуры или давления. Когда измерение будет считано, FIFO автоматически увеличит значение и поместит следующий результат в регистр данных. Флаг будет установлен в регистр состояния FIFO, когда FIFO пуст и все последующие чтения вернут 0x800000. Если FIFO заполнен, бит FIFO_FULL в FIFO Status (FIFO_STS) будет установлен. Если бит INT_FIFO в регистре конфигурации прерываний и FIFO (CFG_REG) установлен, прерывание также будет генерироваться при заполнении FIFO. FIFO остановит запись результатов измерений, когда он будет заполнен.

DPS310 является калиброванным датчиком и содержит калибровочные коэффициенты. Они используются в приложении (например, хост-процессором) для компенсации результатов измерения нелинейностей датчика.

Чтобы достичь более высокой точности, DPS310 будет считывать датчик несколько раз (передискретизация) и объединять показания в один результат. Это увеличивает потребление тока, а также время измерения, уменьшая максимально возможную скорость измерения. Необходимо сбалансировать точность и скорость передачи данных, необходимую для каждого приложения, с допустимым потреблением тока. Точность измерения, скорость и время задаются в регистрах конфигурации давления (PRS_CFG) и конфигурации температуры (TMP_CFG).

Описания регистра содержат информацию о текущем потреблении и возможных комбинациях точности измерения, времени и скорости. Включение измерений температуры позволяет компенсировать понижение температуры при измерении давления.

Скорости этих измерений могут быть установлены независимо, но температурная компенсация более точна, когда измерения температуры и давления взяты вместе. Это уменьшает максимальную скорость измерения давления, так как:

$$V_T * t_T + V_P * t_P < 1 \text{ секунды}$$

Где, V_T – скорость температур, t_T – время температуры, V_P – скорость давления, t_P – время давления.

Настройки измерений и примеры вариантов использования содержат таблицу с примерами сочетаний точности давления и температуры и скоростей для различных

вариантов использования. На рисунке 2.11 ниже описана последовательность измерений температуры и давления в фоновом режиме.

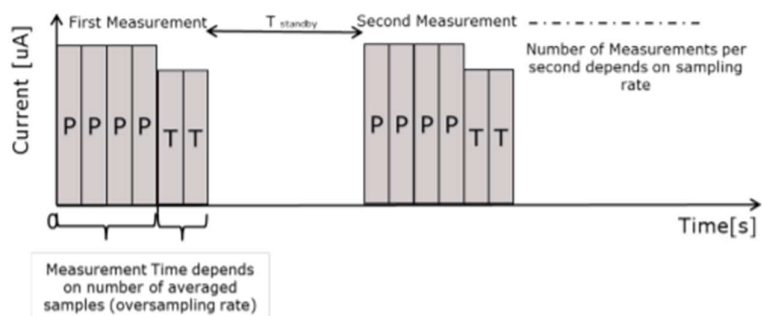


Рисунок 2.11 – Последовательность измерений

2.3.4 Датчик компас

Система обнаружения разливов нефтепродуктов оснащена оптическим оборудованием спектральной и RGB камерой. Камеры располагаются как во внутренней части основного кейса системы, так и во внешней части кейсы. Важно понимать система так же устанавливается на статическую и динамическую платформу. Например, флагшток на буровой установки нефтегазового промысла – это статическая система, где опасные зоны для мониторинга могут выступать постоянно подверженные участки возникновения утечек. А в качестве динамических платформ можно отнести подвижные объекты, которые производят определенную деятельность имеющую цикличность: большая и малая авиация, грузовые суда и т. д.

Оптическая оборудование установленная на статическую или динамическую платформу должна понимать в каких углах и плоскостях находится. Для решения такой задачи нужно правильно и точно определять их.

В данной работе был выбран датчик с возможностью калибровки магнитометра и обладающей достаточной точностью для достоверных получаемых данных – LIS3MDL.

LIS3MDL – это трехосный магнитный датчик со сверхнизким энергопотреблением и высокой производительностью.

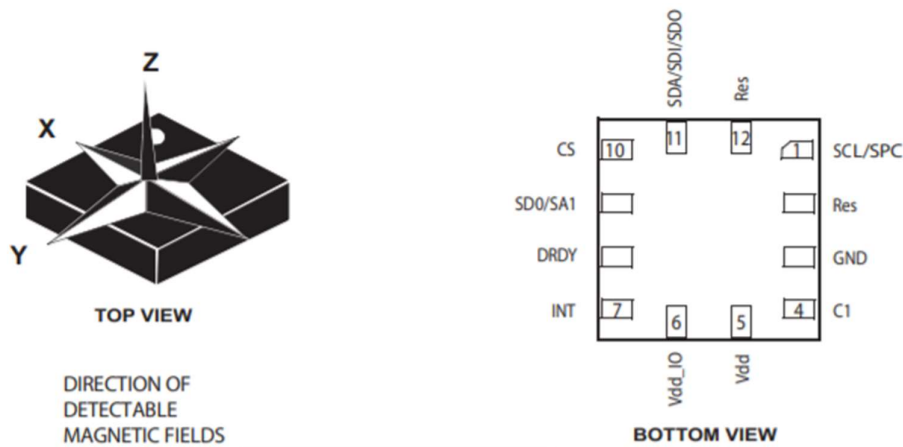


Рисунок 2.12 – Магнитометр LIS3MDL

LIS3MDL имеет выбираемые пользователем полные шкалы $\pm 4 / \pm 8 / \pm 12 / \pm 16$ Гаусс. Возможность самотестирования позволяет пользователю проверить работу датчика в конечном приложении. Устройство может быть сконфигурировано для генерации сигналов прерывания для обнаружения магнитного поля.

LIS3MDL поставляется в небольшом корпусе из тонкой пластиковой сетки (LGA) и гарантированно работает в расширенном диапазоне температур от -40°C до $+85^{\circ}\text{C}$.

LIS3MDL включает интерфейс последовательной шины I2C, который поддерживает стандартный и быстрый режим (100 кГц и 400 кГц) и интерфейс последовательного стандарта SPI.

С шиной I2C связаны два сигнала: последовательная линия синхронизации (SCL) и последовательная линия передачи данных (SDA). Последняя является двунаправленной линией, используемой для отправки и получения данных в / из интерфейса. Обе линии должны быть подключены к Vdd_IO через внешний подтягивающий резистор. Когда автобус свободен, обе линии высокие. Интерфейс I2C соответствует стандартам I2C в быстром режиме (400 кГц), а также нормальному режиму. Когда используется интерфейс I2C, вывод SDO / SA1 должен быть подключен к Vdd_IO или GND.

Так как данная система предполагает, что система может быть разной по установке оптической системы, датчик будет выносным. На рынке присутствует множество готовых решений с уже распаянными микросхемами LIS3MDL. Такие как TROYKA – модуль, POLULO модуль и т.д. И все модули обладают схожими интерфейсами подключения, габаритами, ценой и качеством. Был выбран модуль в исполнении TROYKA – модуль.

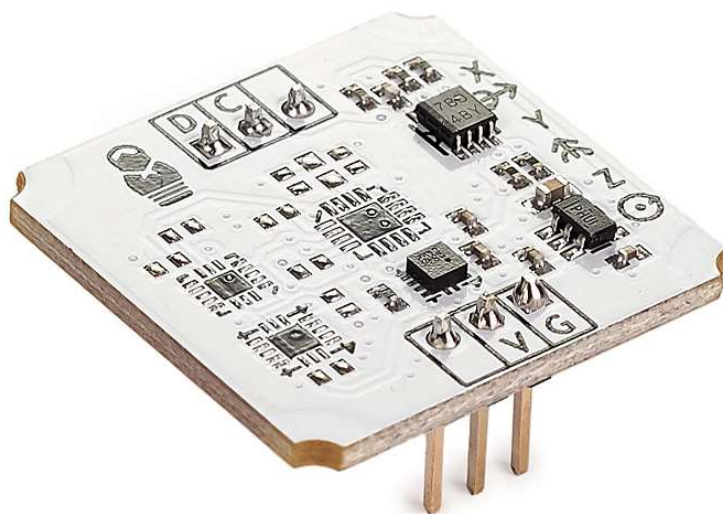


Рисунок 2.13 – Магнитометр/компас TROYKA модуль

2.3.5 Датчик GPS/ГЛОНАСС

Позиционирование в пространстве системы важная роль для обнаружения разливов нефтепродуктов, так как система должна определять свое местоположение и выдавать будущие границы по GPS координатам.

Поэтому для такой задачи был выбран датчик GPS/GLONASS – TROYKA V2 модуль основанный на процессоре NEOWAY G7A.

NEOWAY G7A – это модуль GNSS, который поддерживает BDS B1, GPS L1 и GLONASS L1. Он принимает унифицированный набор микросхем, объединяющий базовую полосу и радиочастоту, и его размеры составляют 10,6 мм x 9,7 мм x 2,2 мм. Это оптимальное навигационное решение для требований высокой чувствительности, низкого энергопотребления и низкой стоимости в различных навигационных устройствах.

Модуль GPS/GLONASS v2 принимает сигналы спутников глобального позиционирования — GPS, GLONASS и Galileo — и рассчитывает свои

географические координаты, скорость перемещения, высоту над уровнем моря и точное локальное время.

Данные передаются на управляющую электронику в текстовом формате NMEA по интерфейсу UART.

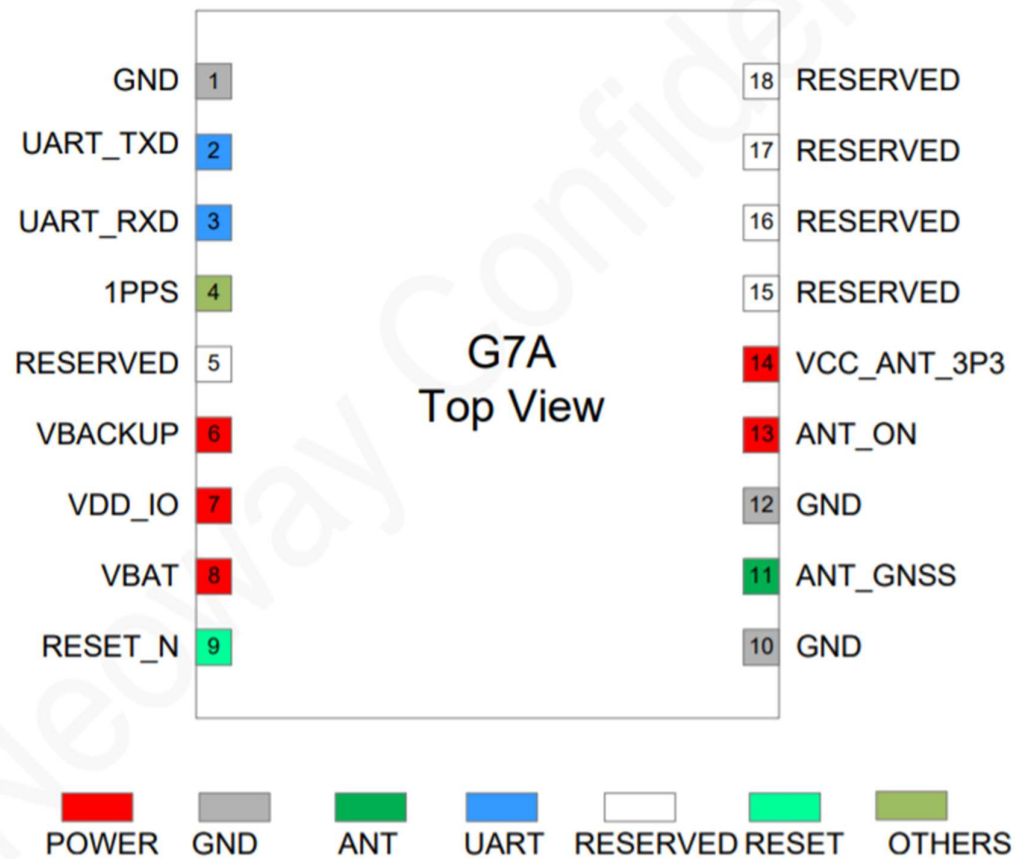


Рисунок 2.14 – Вход/выход микросхемы NEOWAY G7A

Пассивная антенна GPS/GLONAS на керамической подложке необходима для приёма сигнала навигационным приемником GPS/GLONASS.

Данные с GPS-модуля передаются на управляющую электронику в текстовом формате NMEA, состоят из последовательного набора данных, разделенных запятыми. Каждое NMEA сообщение начинается с \$, заканчивается \n (перевод строки) и не может быть длиннее 80 символов. Список сообщений в строке:

- AAM – Прибытие в путевую точку
- ALM – Данные альманаха
- APA – Данные автопилота «А»
- APB – Данные автопилота «В»
- BOD – Азимут на пункт назначения

- DTM – Используемый датум
- GGA – Информация о фиксированном решении
- GLL – Данные широты и долготы
- GSA – Общая информация о спутниках
- GSV – Детальная информация о спутниках
- MSK – Передача управлению базовому приемнику
- MSS – Статус базового приемника
- RMA – Рекомендованный набор данных системы «Loran»
- RMB – Рекомендованный набор навигационных GPS данных
- RMC – Рекомендованный минимальный набор GPS данных
- RTE – Маршрутная информация
- VTG – Вектор движения и скорости
- WCV – Данные скорости вблизи путевой точки
- WPL – Данные путевой точки
- XTC – Ошибка отклонения от трека
- XTE – Измеренная ошибка отклонения от трека
- ZTG – UTC время и оставшееся время до прибытия в точку назначения
- ZDA – Дата и время.

Некоторые из NMEA сообщений могут содержать одинаковые поля данных, либо полностью содержать данные других, меньших по размеру, NMEA сообщений.

GGA - информация о фиксированном решении. Самое популярное и наиболее используемое NMEA сообщение с информацией о текущем фиксированном решении – горизонтальные координаты, значение высоты, количество используемых спутников и тип решения.

\$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47

- GGA – NMEA Заголовок
- 123519 –UTC время 12:35:19
- 4807.038, N – Широта, 48 градусов 7.038 минуты северной широты
- 01131.000, E – Долгота, 11 градусов 31.000 минуты восточной долготы
- 1 – тип решение, StandAlone решение
 - 0 – нет решения,
 - 1 – StandAlone,
 - 2 – DGPS,

- 3 – PPS,
- 4 – фиксированный RTK,
- 5 – не фиксированный RTK,
- 6 – использование данных инерциальных систем,
- 7 – ручной режим,
- 8 – режим симуляции
- 08 – количество используемых спутников
- 0.9 – геометрический фактор, HDOP
- 545.4, М – высота над уровнем моря в метрах
- 46.9, М – высота геоида над эллипсоидом WGS 84
- [пустое поле] – время прошедшее с момента получения последней DGPS поправки. Заполняется при активизации DGPS режима
- [пустое поле] – идентификационный номер базовой станции. Заполняется при активизации DGPS режима.

RMC — рекомендованный минимальный набор GPS данных. Это NMEA сообщение содержит весь набор, так называемых «PVT» данных. «PVT» — общепринятое сокращение от «position, velocity, time» (позиция, скорость, время).

\$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W*6A

- RMC – NMEA заголовок
- 123419 – UTC время, 12:34:59
- A – статус (A- активный, V- игнорировать)
- 4807.038,N – Широта, 48 градусов 07.038 минут северной широты
- 01131.000,E – Долгота, 11 градусов 31.000 минута восточной долготы
- 022.4 – Скорость, в узлах
- 084.4 – Направление движения, в градусах
- 230394 – Дата, 23 марта 1994 года
- 003.1,W – Магнитные вариации

Характеристики модуля ТРОЙКА GPS/GLONASS V2:

- Чипсет: Neoway G7
- Навигационные системы: GPS, GLONASS, BDS
- Рабочие частоты:
 - GPS L1: 1575.42 МГц
 - GLONASS L1: 1602,5625 МГц

- BDS L1: 1561,098 МГц
- Протоколы передачи данных: NMEA
- Скорость передачи данных по умолчанию: 9600 бод
- Размеры встроенной антенны: 40×40 мм
- Поляризация: RHCP (правая круговая поляризация)
- Погрешность позиционирования: менее 3 м
- Горячий старт: 1 с
- Холодный старт: 28 с
- Рабочее напряжение: 3,3–5 В
- Потребляемый ток: до 25 мА
- Габариты: 25,4×25,4 мм

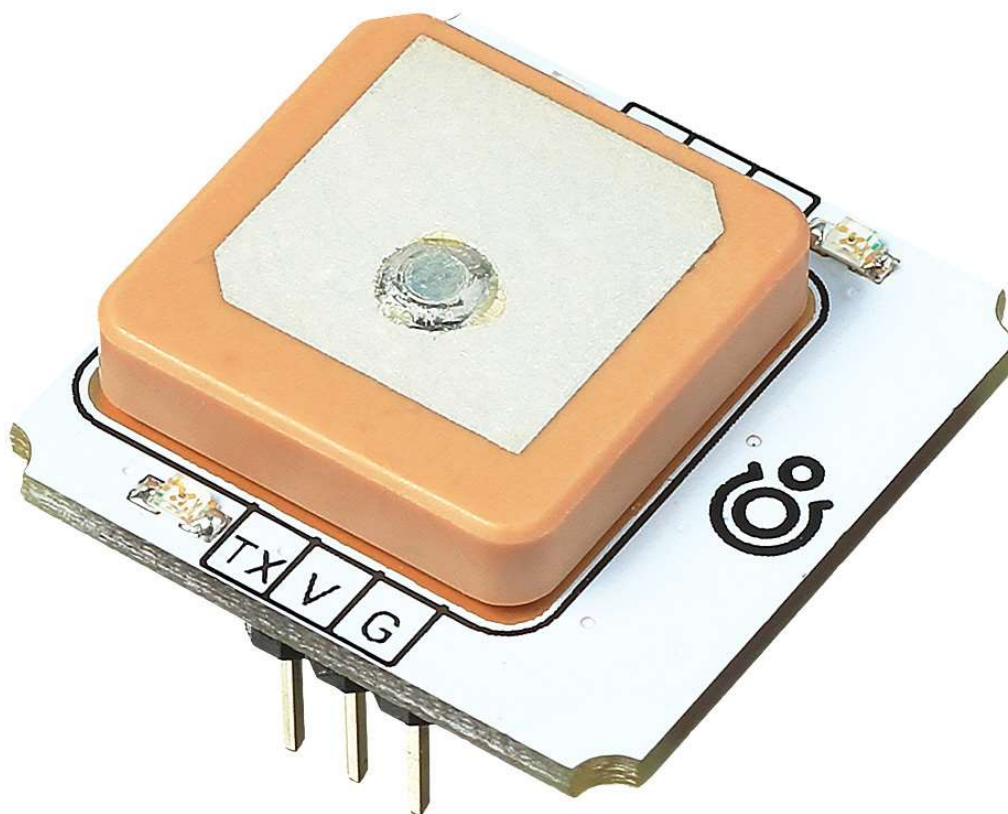


Рисунок 2.15 – Модуль TROYKA GPS/GLONASS V2\

2.3.6 Спектральная и RGB камера

Основным элементом сенсоров для обнаружения физических и химических параметров нефтепродуктов является оптическая система. В оптическую систему входит два оборудования это спектральная и RGB камера.

RGB камера довольно проста в эксплуатации и обработки изображения. Такая камера должна обладать оптимальными характеристиками для внедрения в систему обнаружения. Так как система может находиться на мобильной платформе в виде БПЛА или же суден, при выборе камеры важно учитывать разрешение съемки.

В работе в качестве RGB камеры была выбрана камера iXM – 100. Являясь крупным игроком на переднем крае аэрофотосъемки, Phase One Industrial разработала серию iXM - революционные и уникально разработанные камеры для приложений для создания изображений БПЛА, которые переопределяют границы технологических инноваций.

Технические особенности камеры iMX – 100 представлены ниже:

- Высокое разрешение с количеством пикселей:
 - 11664x8750 для iXM 100MP
 - 8280x6208 для iXM 50MP
- 3,76 мкм для iXM 100MP
- Датчик среднего формата с технологией задней подсветки
- Высокая скорость захвата 3 кадра в секунду для iXM 100MP
- Выбор из пяти специально разработанных линз RSM
- Простая интеграция с различными платформами БПЛА - с последовательным управлением камерой и выходом HDMI, включая состояние камеры и данные изображения в оверлее
- USB-C и 10G Ethernet для универсальности подключения к крупным БЛА и самолетам
- Максимальная мощность потребления 16 Ватт
- Входное напряжение 16 – 30 Вольт
- Температурный диапазон работы от – 10 °C до 50 °C



Рисунок 2.16 – RGB камера iMX – 100

Что касательно спектральной камеры, главной характеристикой является диапазон спектрального излучения, который он может принимать, и массогабаритные характеристики.

Спектральное излучение нефтепродуктов лежит в диапазоне от 400 нм до 1000 нм. В ходе анализа оптических систем была выбрана гиперспектральная камера MV1-D2048x1088-HS01-96-G2.



Рисунок 2.17 – Гиперспектральная камера MV1-D2048x1088-HS01-96-G2

В камере MV1-D2048x1088-HS01-G2 используется клиновой сканер IMEC CMV2K-LS100-NIR для линейного сканирования IMEC, основанный на CMOS-датчике CMOSIS CMV2000. Датчик имеет в спектральной области от 600 нм до 1000 нм 100 полос пропускания с полувысотой по всей ширине (FWHM) от 10 до 12 нм.

Камеры ориентированы на требовательные приложения для гиперспектральной визуализации. Низкая полная лунка и общая заслонка датчика делают возможными высокоскоростные применения. Камера имеет интерфейс GigE (GigEVision).

Технические параметры камеры указаны ниже:

- IMEC CMV2K-LS100-NIR CMOS датчик изображения
- разрешение 2048 x 1088 пикселей
- Хороший NIR спектральный отклик
- Подходит для гиперспектральных приложений
- До 42 кадров в секунду при полном разрешении
- Глобальный затвор
- 100 полос пропускания от 600 до 1000 нм
- Расширенные функции датчика и камеры
- Биннинг для предварительной обработки данных
- Разрешение до 10 бит в оттенках серого
- Доступны решения Boardlevel и OEM.
- интерфейс GigEVision

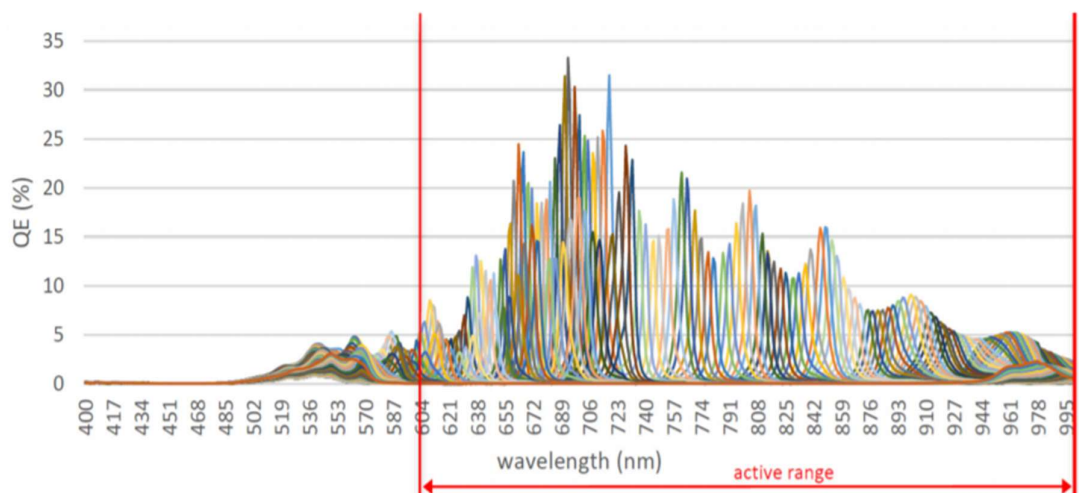


Рисунок 2.18 – Полосы пропускания спектрального излучения камеры

Блок биннинга представляет особый интерес для камеры MV1-D2048x1088-NS01-96-G2-10. Преимуществом использования вертикального биннинга является высокая частота кадров и наличие только одного значения на полосу для каждого столбца.

Есть возможности увеличить частоту кадров еще больше:

- Функция «Несколько областей интереса (MROI)» может использоваться для выбора только интересующих полос. Считывание меньшего количества строк увеличивает частоту кадров во всех форматах пикселей.
- Горизонтальное разрешение можно уменьшить, выбрав меньшую ширину или используя горизонтальное группирование. Обратите внимание, что это не увеличивает частоту кадров для формата пикселей Mono8, если используется $\text{BinningVertical} = 8$, поскольку датчик уже считывается на полной скорости. Во всех других форматах пикселей это увеличивает частоту кадров.

2.4 Разработка принципиальной электрической схемы

Исходя структурной схемы микропроцессорной системы обнаружения разливов нефтепродуктов и подбора необходимой компонентной базы, была разработана принципиальная схема в программе SPlane 7 и КОМПАС 3D.

При разработке принципиальной схемы учитывалось как компоненты будут между собой связываться, а именно выбраны оптимально подходящие интерфейсы связи. Выше в разделе «Технологии передачи данных» были указаны основные методы коммуникации микропроцессоров, сенсоров и т.д.

Для низкого уровня интерфейсы коммуникации были выбраны I2C для датчика LIS3MDL (компас) и DPS310 (барометр), а для NEOWAY G7A (GPS/GLONASS) был выбран интерфейс UART, так как микросхема имеет возможность взаимодействовать только по этому интерфейсу.

Из-за занятого уже интерфейса UART для NEOWAY G7A, который в основном на микроконтроллерах бывает один. А связь между высоким и низким уровнем осуществляется тоже по интерфейсу UART. Была выбрана плата низкого уровня TEENSY 3.6 имеющий 3 интерфейса UART и с возможностью параллельной работы их.

Датчики ориентации в пространстве барометр, GPS/GLONASS и компас выдают метаданные на микроконтроллер низкого уровня, а в свою очередь он производит калибровку сенсоров, обработку полученных данных, формирует в пакет и отправляет уже на Мини – ПК (высокий уровень).

Высокий уровень, а именно Мини – ПК NVIDIA JETSON AGX XAVIER производит анализ поступающего видеопотока с камер и привязывает метаданные к этому потоку. Мини – ПК соединяется по каналу Ethernet по протоколу GigE.

На основе этих данных была разработана принципиальная схема исполнительного устройства.

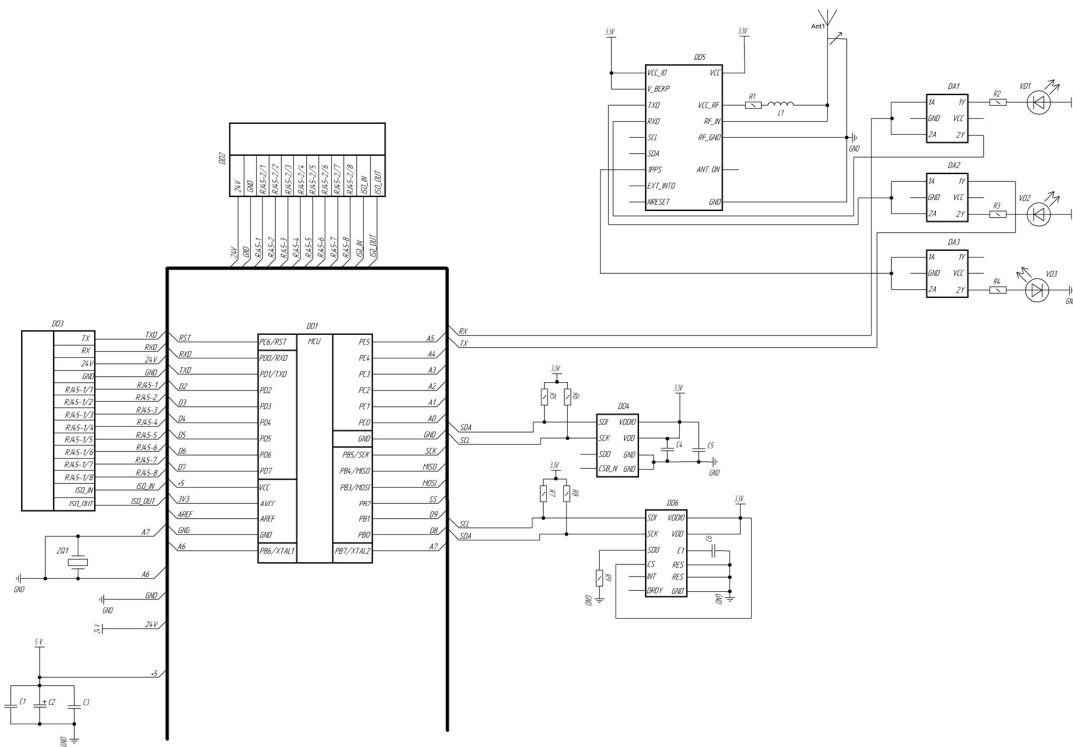


Рисунок 2. 19 – Принципиальная электрическая схема системы

Входы SDA и SCL отвечает за подключение к плате низкого уровня датчиков барометра и компаса по интерфейсу I2C, TX2 и RX2 отвечает за подключение к плате низкого уровня GPS/GLONASS по интерфейсу UART1, а контакты TX1 и RX1 за подключение низкого уровня с высоким уровнем. Контакты RJ45 необходимы для подключения камер по интерфейсу GigE.

2.5 Проектирование печатной платы низкого уровня

На основе разработанной принципиальной схемы была изготовлена печатная плата ШИЛД для низкого уровня в программах DIPTRACE и КОМПАС 3D. Были разработаны библиотеки компонентов модулей.

Данная плата изображена в ПРИЛОЖЕНИИ В графической части работы. Плата включает в себя микроконтроллер TEENSY 3.6, модули TROYKA GPS/GLONASS V2, TROYKA Магнитометр/Компас и S2GO PRESSURE DPS310 и разъемы для подключения к Мини – ПК. На плате также разведены пины для дополнительных устройств.

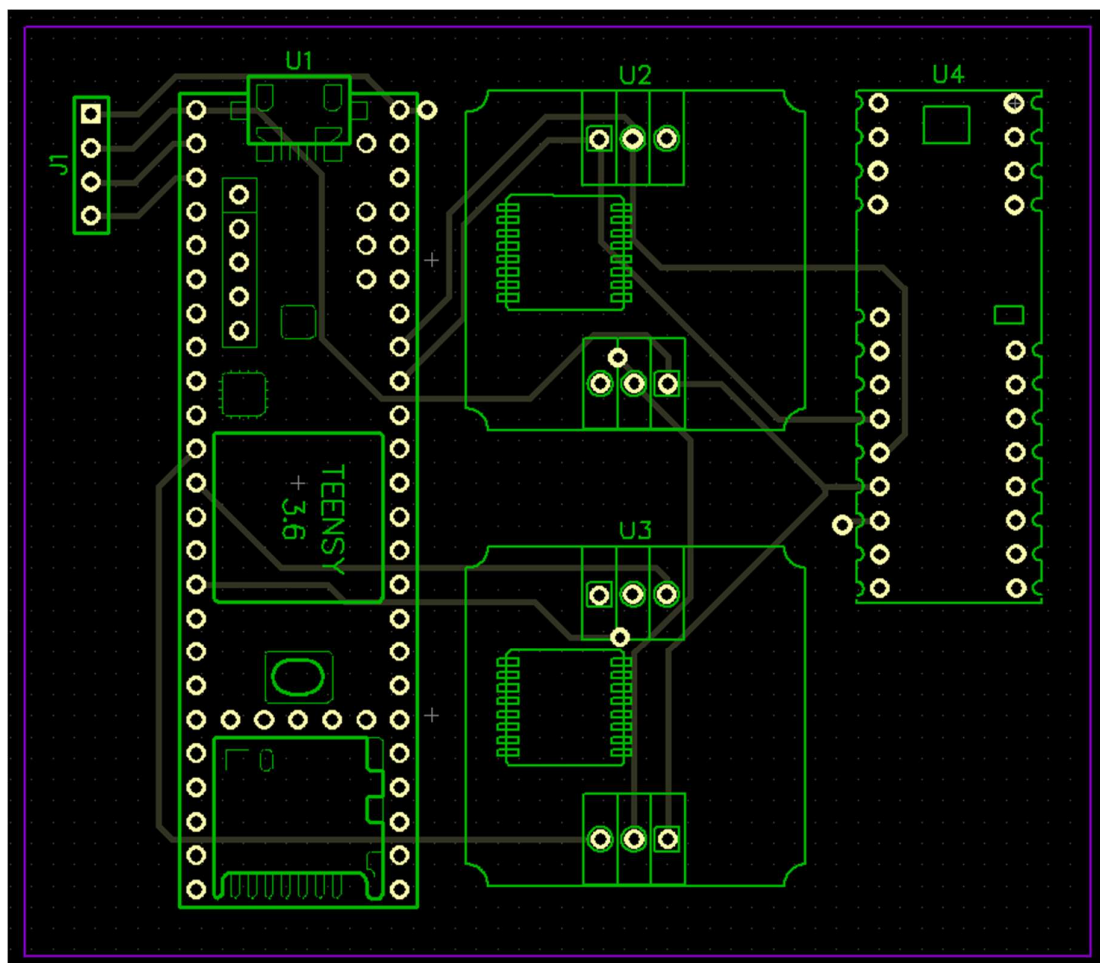


Рисунок 2.20 – Печатная плата низкого уровня верхний слой

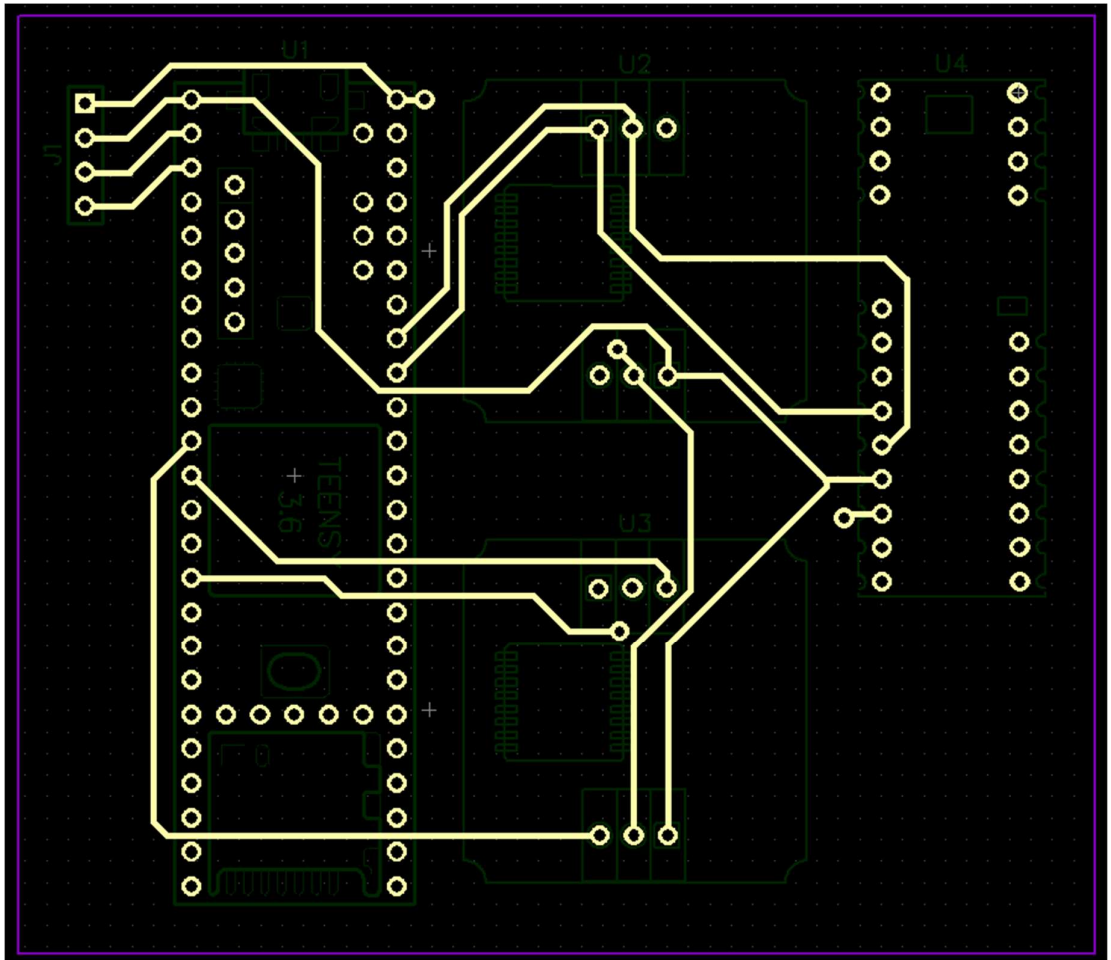


Рисунок 2.20 – Печатная плата низкого уровня нижний слой

3 ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

3.1 Исследование загруженности процессора Jetson AGX

Xavier

Загруженность процессора микропроцессорной техники, значительно сказывается на работе систем отвечающие за решение важных задач в производстве, быту и т. д.

Важно понимать, что при загруженности процессора возникает риски возникновения отказа системы и приводящая к нештатным аварийным ситуациям как на производстве, так и близлежащих зон.

Разработка микропроцессорной системы обнаружения разливов нефтепродуктов может встраиваться как на мобильную платформу, такие как малая авиация, судна и т.д., так и устанавливаться на производстве в местах подверженные риску.

Поэтому, чтобы быть уверенным в выборе оптимальной алгоритма распознавания объектов и оборудования который будет производить решение данной задачи. Был произведен расчет каждого метода анализа распознавания объекта и его загруженности, время задержки обработки получаемых данных и построения вольтамперной характеристики при каждом методе.

Выше, в главе 2 был описал Мини – ПК NVIDIA Jetson AGX Xavier , который отвечает заданным характеристикам производительности и со способностью работай с машинным обучением.

Рассмотрим основные методы и инструменты работы с машинным обучением, а именно:

- ResNet-18 FCN;
- ResNet-50;
- VGG19;
- GoogleNet;
- AlexNet.

На основе этих методов будут проводится измерения общего энергопотребления модуля и энергоэффективности включают использование CPU, GPU, DLA, памяти, разной мощности SoC, входов / выходов и потерь эффективности регулятора на всех шинах. Потребляемая мощность измерялась с помощью INA контроллеров напряжения и тока на борту модуля.

Оценки будущей производительности, включая усовершенствования программного обеспечения, такие как поддержка INT8 для DLA и дополнительные оптимизации GPU, предоставлены для различных конфигураций сети. Будущие оценки производительности предполагают одновременное использование графического процессора (INT8) и двух DLA (INT8).

3.1.1 ResNet – 18 FCN

ResNet-18 - это сверточная нейронная сеть, которая имеет глубину 18 слоев. Вы можете загрузить предварительно обученную версию сети, обученную более чем миллиону изображений, из базы данных ImageNet. Предварительно обученная сеть может классифицировать изображения на 1000 категорий объектов, таких как клавиатура, мышь, карандаш и многие животные. В результате сеть изучила богатые возможности представлений для широкого спектра изображений. Сеть имеет входной размер изображения 224 на 224.

ImageNet — это набор данных миллионов помеченных изображений с высоким разрешением, относящихся примерно к 22 тысячам категорий. Изображения были собраны из Интернета и помечены людьми с помощью краудсорсинга.

Предварительно обученная модель была предварительно обучена на наборе данных и содержит веса и отклонения, которые представляют особенности того набора данных, на котором она была обучена. Изученные функции часто переносятся на другие данные. Например, модель, обученная на большом наборе данных изображений птиц, будет содержать изученные особенности, такие как ребра или горизонтальные линии, которые вы могли бы перенести в свой набор данных.

Предварительно обученные модели выгодны для нас по многим причинам. Используя предварительно обученную модель, вы экономите время. Кто-то уже потратил время и вычислил ресурсы для изучения множества функций, и ваша модель, вероятно, выиграет от этого.

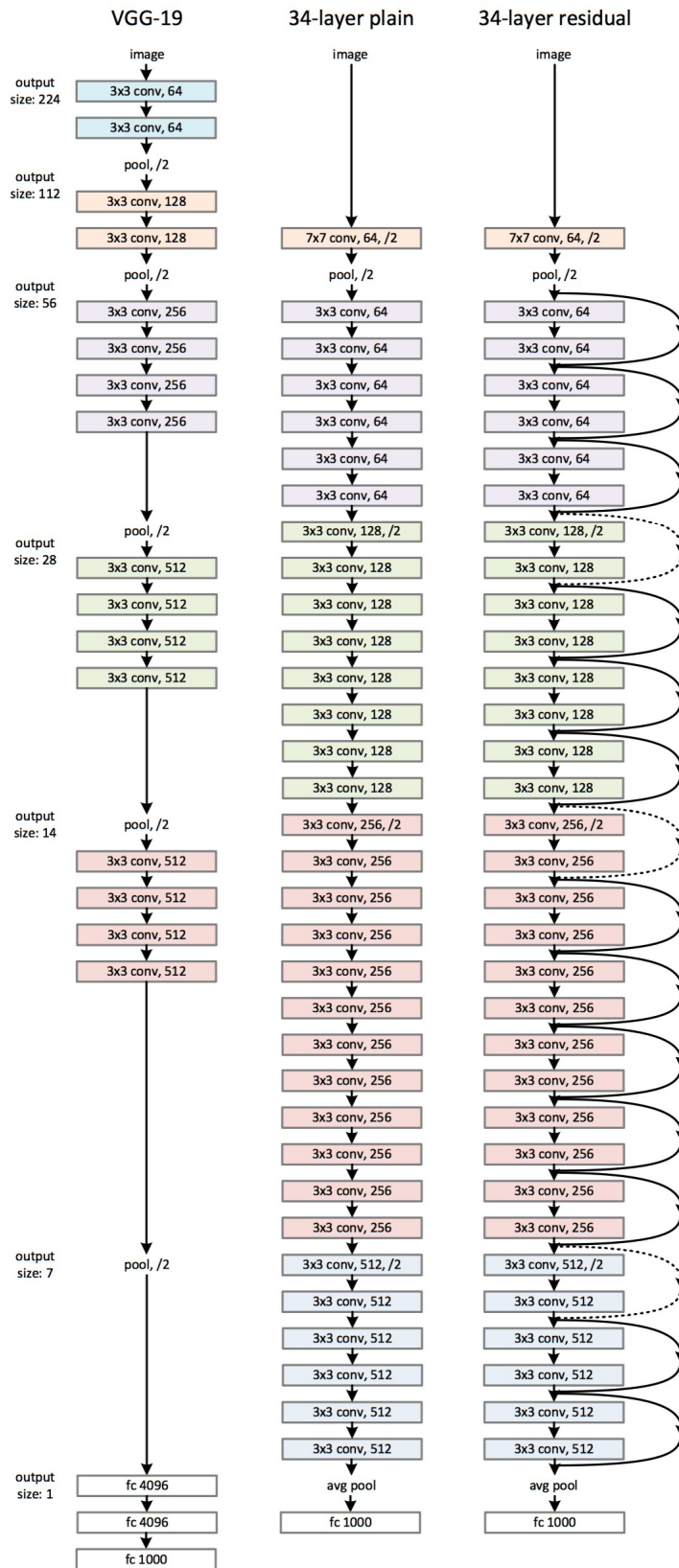


Рисунок 3.1 – Пример сетевой архитектуры для ImageNet

Был выполнен анализ на производительность и энергопотребления в двух режимах: 15 Вт(HD) и MAX – IN (HD). На рисунках 3.2, 3.3 и 3.4 показаны результаты исследования при загрузенности процессора партиями в режиме 15 Вт, состоящими из 1, 2, 4, 16 и 32 кадров изображения, т. е. видеопоток с разной частотой кадров.

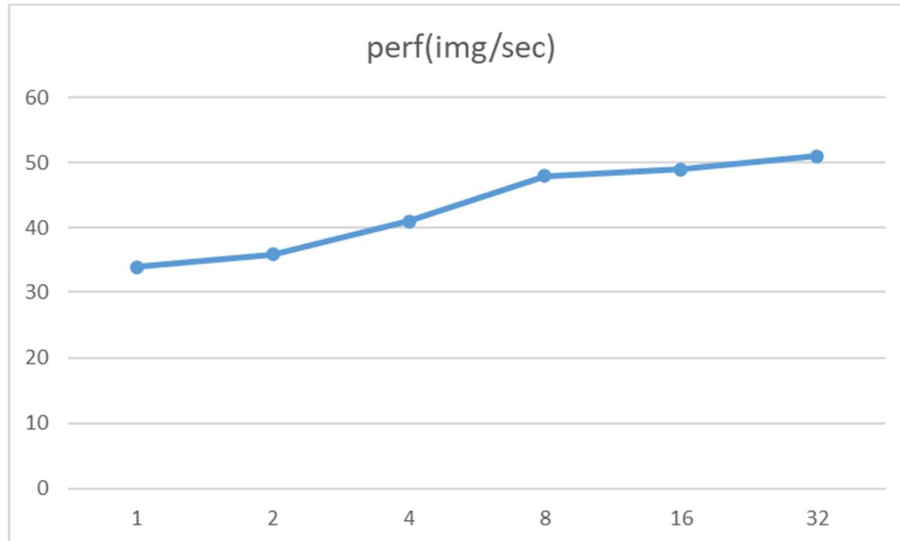


Рисунок 3.2 – График производительности изображения в секунду метода ResNet-18 на режиме 15 Вт

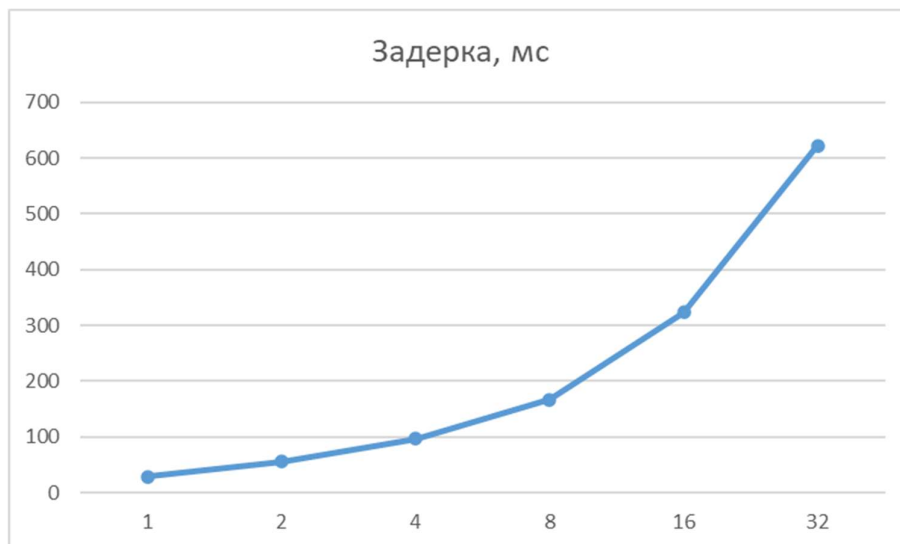


Рисунок 3.3 – График задержки выполнения партии потока метода ResNet-18 на режиме 15 Вт

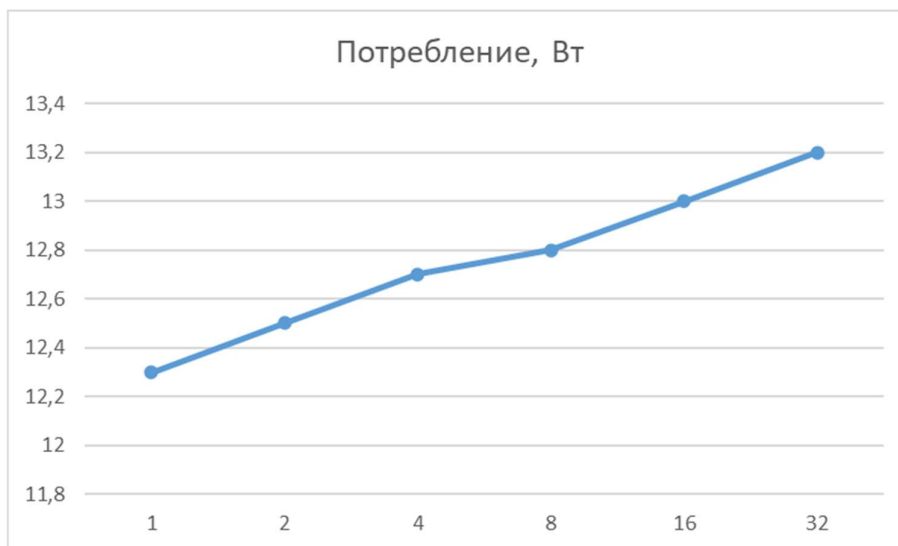


Рисунок 3.4 – График энергопотребления метода ResNet-18 на режиме 15 Вт

На рисунках 3.5, 3.6 и 3.7 показаны результаты исследования при загрузке процессора партиями в режиме MAX - IN, состоящими также из 1, 2, 4, 16 и 32 кадров изображения.

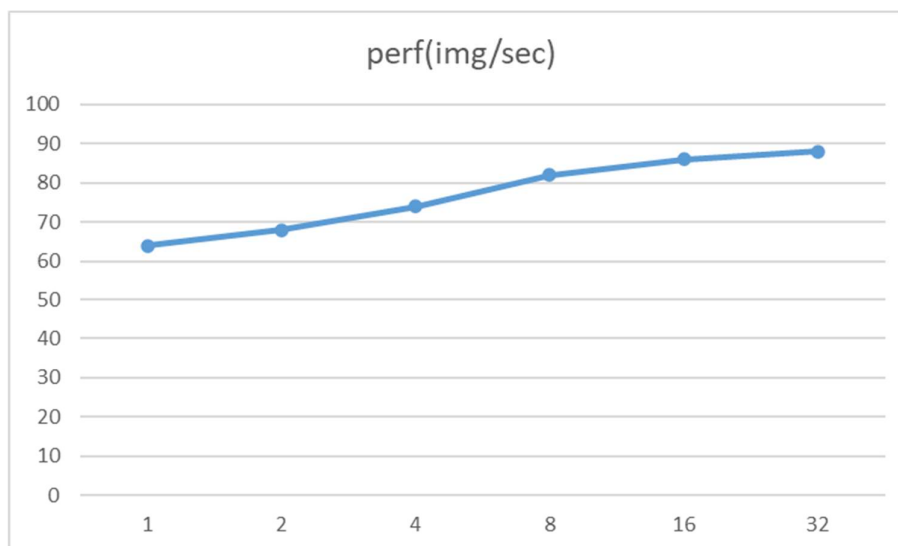


Рисунок 3.5 – График производительности изображения в секунду метода ResNet-18 на режиме MAX – IN

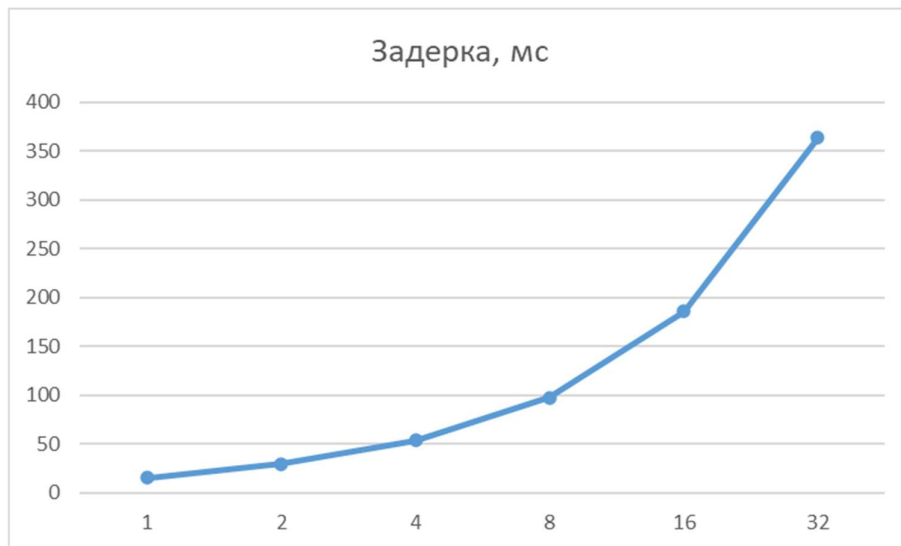


Рисунок 3.6 – График задержки выполнения партии потока метода ResNet-18 на режиме MAX – IN



Рисунок 3.7 – График энергопотребления метода ResNet-18 на режиме MAX – IN

3.1.2 ResNet-50

ResNet-50 – это сверточная нейронная сеть глубиной 50 слоев. Вы можете загрузить предварительно обученную версию сети, обученную более чем миллиону изображений, из базы данных ImageNet [1]. Предварительно обученная сеть может классифицировать изображения на 1000 категорий объектов, таких как клавиатура, мышь, карандаш и многие животные.

В случае ResNet -50 архитектура изменяется. Каждый 3-слойный блок заменяется в 34-слойной сети этим 3-слойным узким местом, в результате получается 50-слойная ResNet.

Был выполнен анализ на производительность и энергопотребления в двух режимах: 15 Вт(HD) и MAX – IN (HD). На рисунках 3.8, 3.9 и 3.10 показаны результаты исследования при загрузенности процессора партиями в режиме 15 Вт, состоящими из 1, 2, 4, 16, 32, 64 и 128 кадров изображения, т. е. видеопоток с разной частотой кадров.

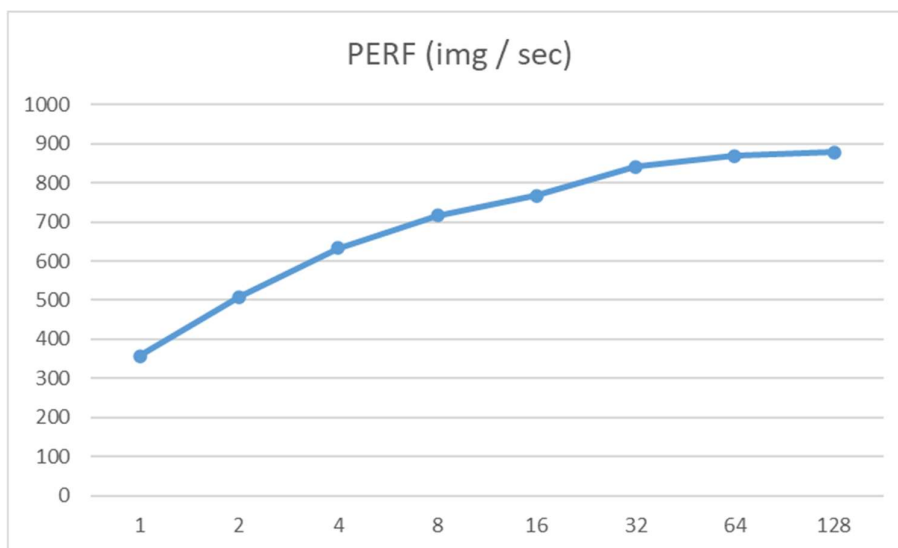


Рисунок 3.8 – График производительности изображения в секунду метода ResNet-50 на режиме 15 Вт

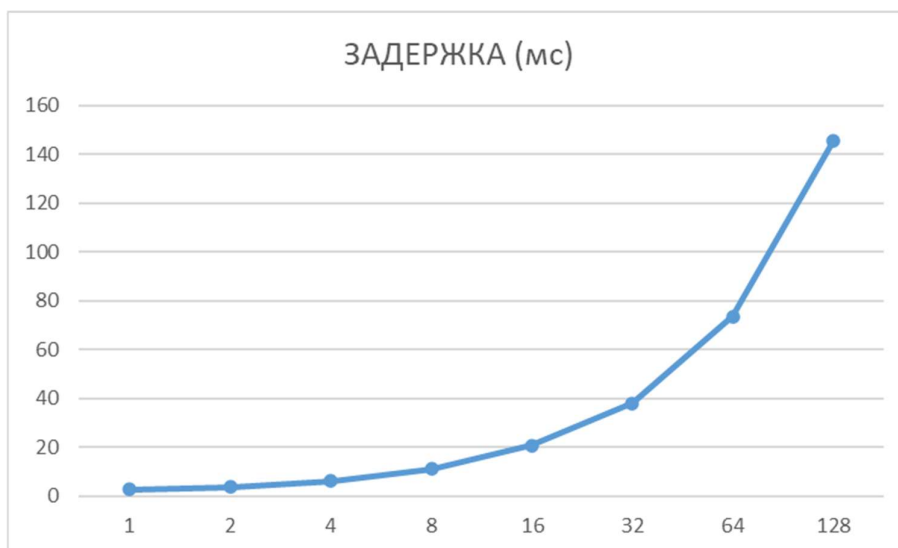


Рисунок 3.9 – График задержки выполнения партии потока метода ResNet-50 на режиме 15 Вт

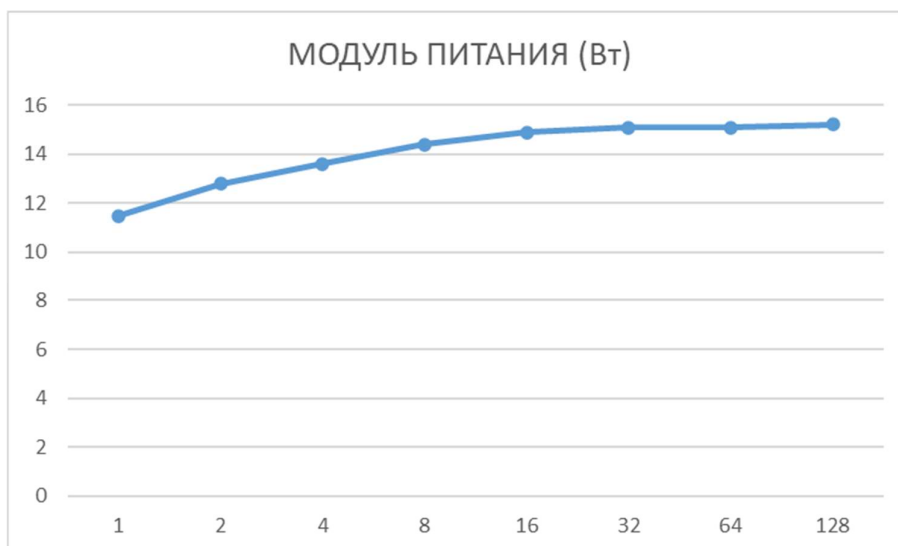


Рисунок 3.10 – График энергопотребления метода ResNet-50 на режиме 15 Вт

На рисунках 3.11, 3.12 и 3.13 показаны результаты исследования при загрузенности процессора партиями в режиме MAX - IN, состоящими также из 1, 2, 4, 16, 32, 64 и 128 кадров изображения.

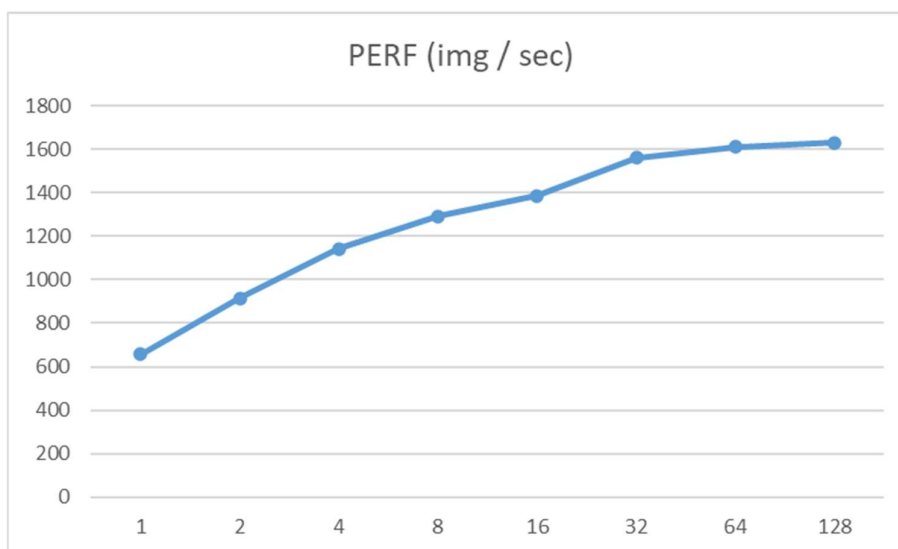


Рисунок 3.11 – График производительности изображения в секунду метода ResNet-50 на режиме MAX – IN

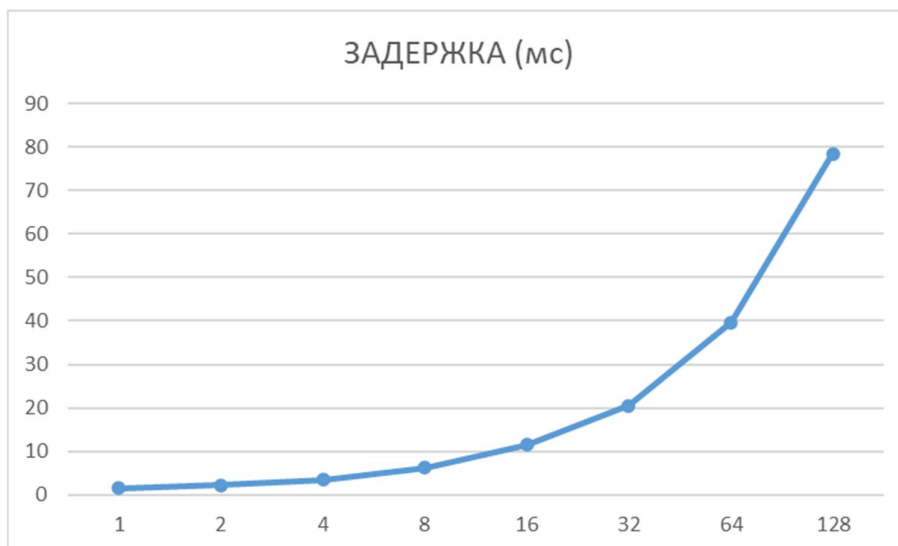


Рисунок 3.12 – График задержки выполнения партии потока метода ResNet-50 на режиме MAX – IN



Рисунок 3.13 – График энергопотребления метода ResNet-50 на режиме MAX – IN

3.1.3 VGG19

«Нейронная сеть VGG-19» определяется как нейронная сеть такого типа, которая также специально обучается на более чем миллионе изображений из базы данных «ImageNet», но разница между «Нейронной сетью VGG-19» и «Нейронной сетью VGG-16» в том, что этот тип сети имеет глубину 19 слоев, а сеть этого типа имеет соответственно 16 уровней.

VGG принимает изображение RGB 224x224 пикселей. Для конкурса ImageNet авторы обрезали центральный патч 224x224 в каждом изображении, чтобы сохранить постоянный размер входного изображения.

Сверточные слои в VGG используют очень маленькое восприимчивое поле (3x3, наименьший возможный размер, который все еще захватывает влево / вправо и вверх / вниз). Существуют также сверточные фильтры 1x1, которые действуют как линейное преобразование входа, за которым следует модуль ReLU. Шаг свертки фиксируется на 1 пикселе, так что пространственное разрешение сохраняется после свертки.

Полностью связанные слои VGG имеет три полностью связанных слоя: первые два имеют 4096 каналов каждый, а третий имеет 1000 каналов, по 1 для каждого класса.

Все скрытые слои VGG используют ReLU (огромное нововведение от AlexNet, которое сокращает время обучения). VGG обычно не использует Local Response Normalization (LRN), поскольку LRN увеличивает потребление памяти и время обучения без особого увеличения точности.

VGG – это инновационная модель распознавания объектов, которая поддерживает до 19 слоев. Созданный как глубокий CNN, VGG также превосходит базовые показатели по многим задачам и наборам данных за пределами ImageNet. В настоящее время VGG по-прежнему является одной из наиболее часто используемых архитектур распознавания изображений.

Был выполнен анализ на производительность и энергопотребления в двух режимах: 15 Вт(HD) и MAX – IN (HD). На рисунках 3.14, 3.15 и 3.16 показаны результаты исследования при загрузке процессора партиями в режиме 15 Вт, состоящими из 1, 2, 4, 16, 32, 64 и 128 кадров изображения, т. е. видеопоток с разной частотой кадров.

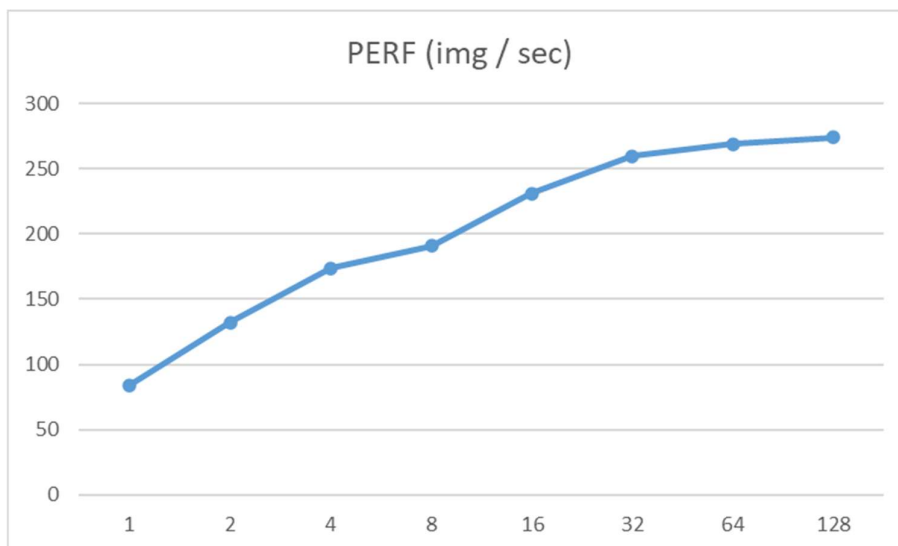


Рисунок 3.14 – График производительности изображения в секунду метода VGG19 на режиме 15 Вт

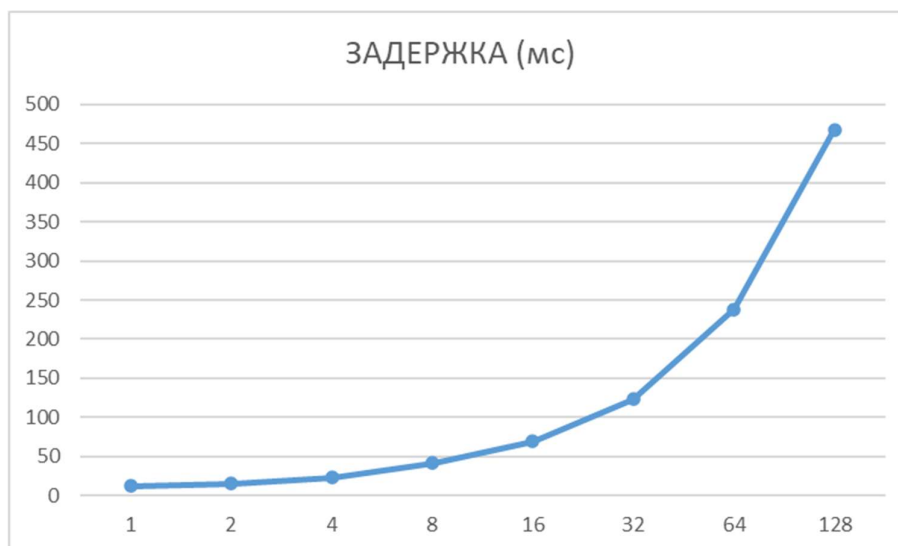


Рисунок 3.15 – График задержки выполнения партии потока метода VGG19 на режиме 15 Вт



Рисунок 3.16 – График энергопотребления метода VGG19 на режиме 15 Вт

На рисунках 3.17, 3.18 и 3.19 показаны результаты исследования при загрузке процессора партиями в режиме MAX - IN, состоящими также из 1, 2, 4, 16, 32, 64 и 128 кадров изображения.

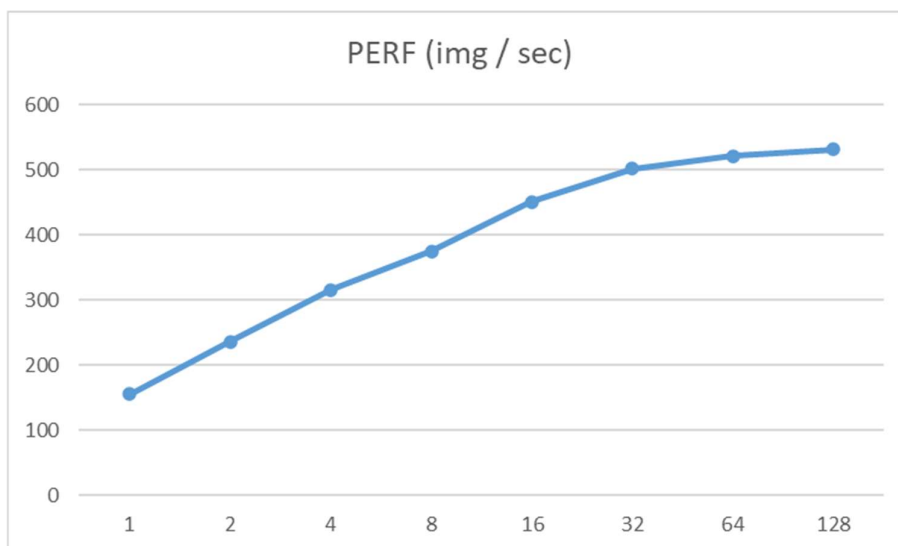


Рисунок 3.17 – График производительности изображения в секунду метода VGG19 на режиме MAX – IN

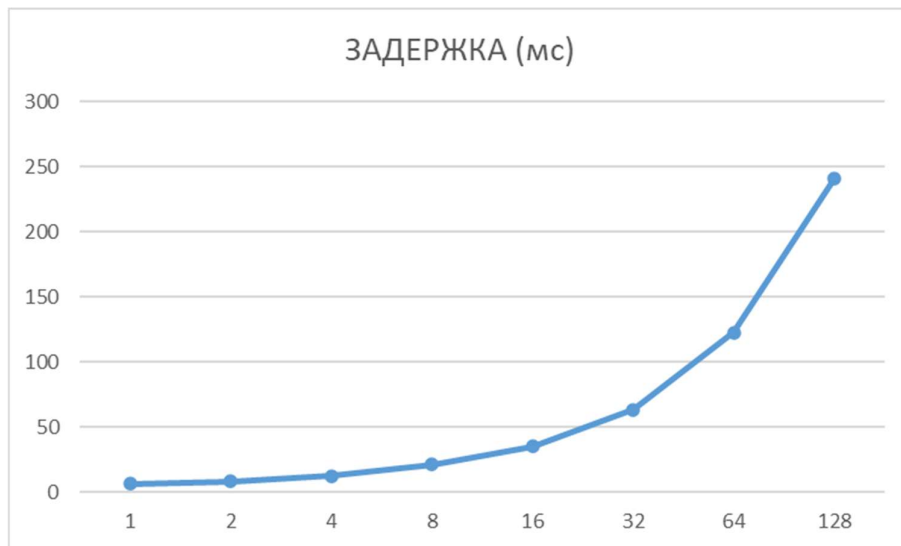


Рисунок 3.18 – График задержки выполнения партии потока метода VGG19 на режиме MAX – IN

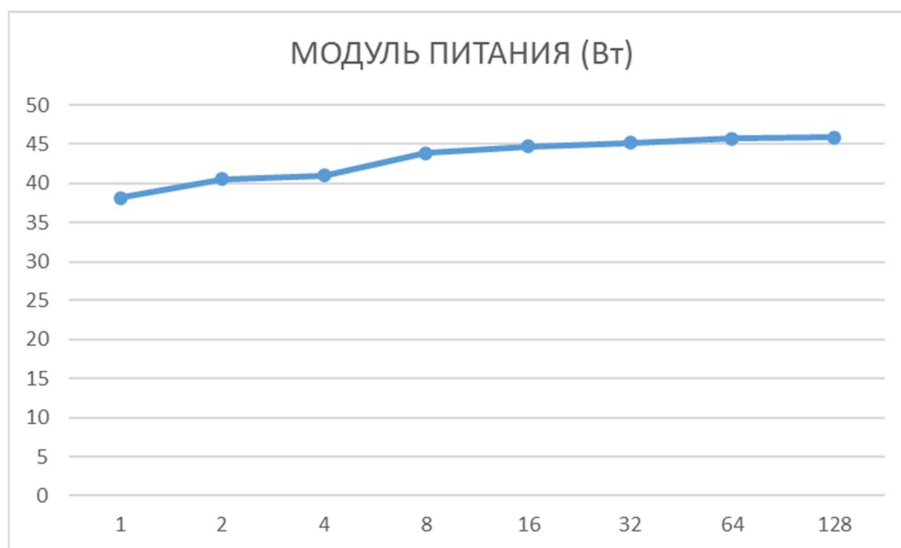


Рисунок 3.19 – График энергопотребления метода VGG19 на режиме MAX – IN

3.1.4 GoogleNet

GoogLeNet – это сверточная нейронная сеть с глубиной 22 слоя. Вы можете загрузить предварительно обученную версию сети, обученную в наборах данных ImageNet [1] или Places365 [2] [3]. Сеть, обученная в ImageNet, классифицирует изображения на 1000 категорий объектов, таких как клавиатура, мышь, карандаш и многие животные. Сеть, обученная на Places365, аналогична сети, обученной на ImageNet, но классифицирует изображения на 365 различных категорий мест, таких как поле, парк, взлетно-посадочная полоса и вестибюль.

Был выполнен анализ на производительность и энергопотребления в двух режимах: 15 Вт (HD) и MAX – IN (HD). На рисунках 3.20, 3.21 и 3.22 показаны результаты исследования при загрузенности процессора партиями в режиме 15 Вт, состоящими из 1, 2, 4, 16, 32, 64 и 128 кадров изображения, т. е. видеопоток с разной частотой кадров.

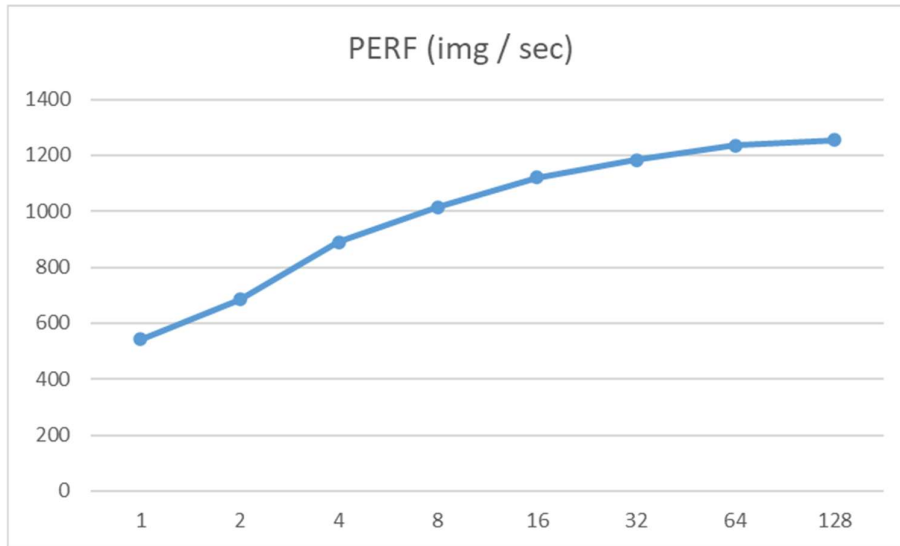


Рисунок 3.20 – График производительности изображения в секунду метода GoogleNet на режиме 15 Вт

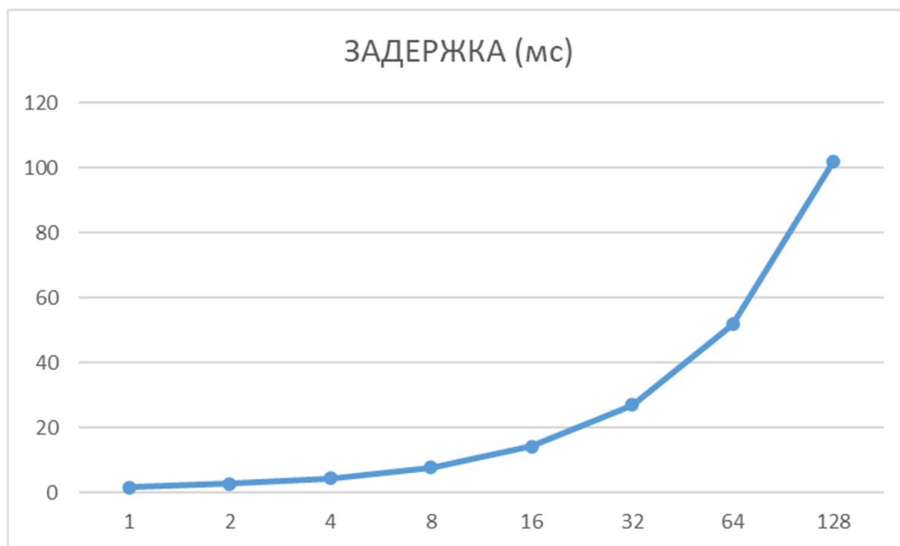


Рисунок 3.21 – График задержки выполнения партии потока метода GoogleNet на режиме 15 Вт

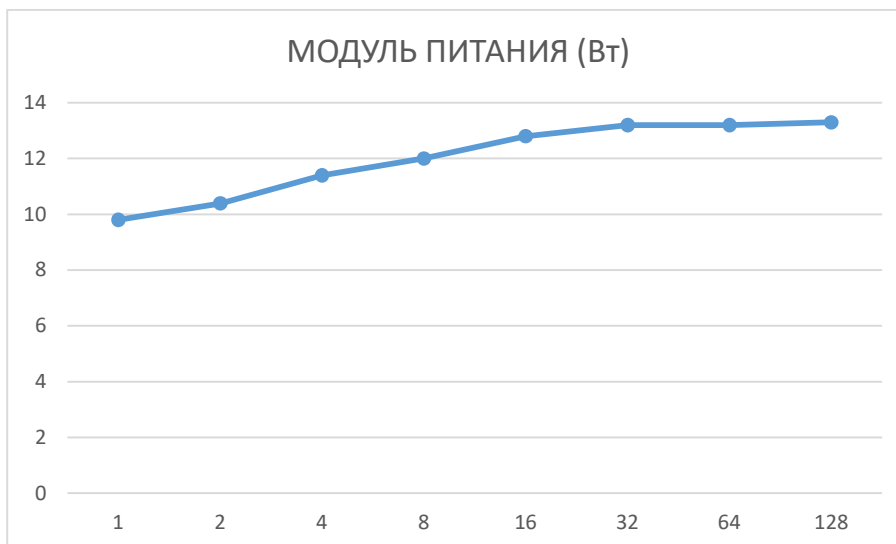


Рисунок 3.22 – График энергопотребления метода GoogleNet на режиме 15 Вт

На рисунках 3.23, 3.24 и 3.25 показаны результаты исследования при загрузенности процессора партиями в режиме MAX - IN, состоящими также из 1, 2, 4, 16, 32, 64 и 128 кадров изображения.

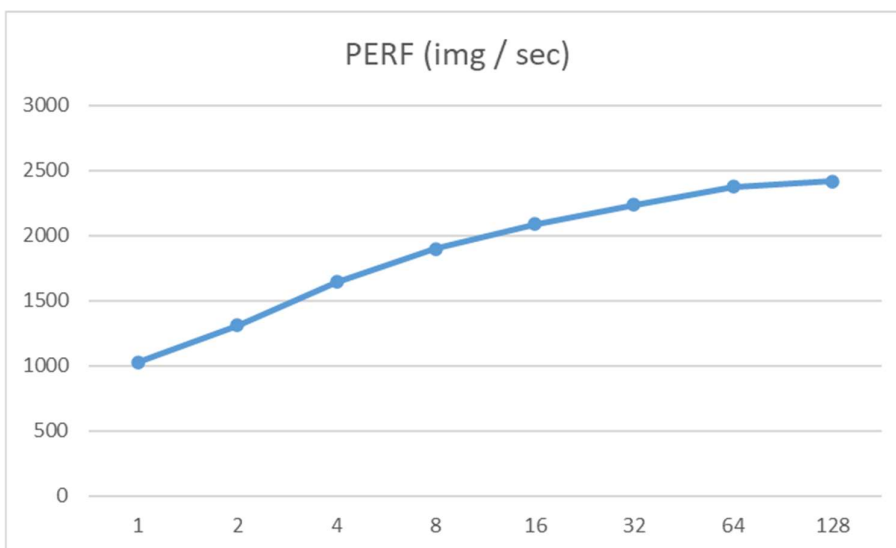


Рисунок 3.23 – График производительности изображения в секунду метода GoogleNet на режиме MAX – IN

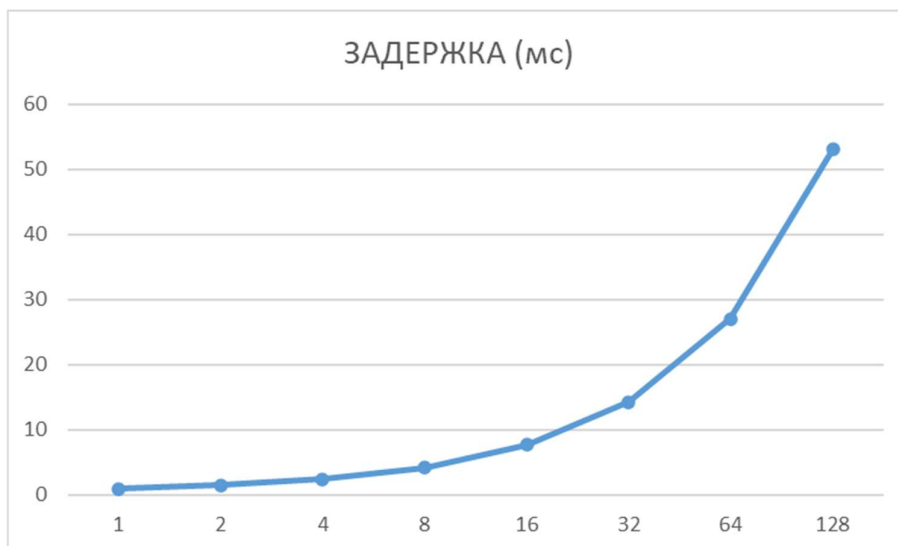


Рисунок 3.24 – График задержки выполнения партии потока метода GoogleNet на режиме MAX – IN



Рисунок 3.25 – График энергопотребления метода GoogleNet на режиме MAX – IN

3.1.5 AlexNet

AlexNet – это сверточная нейронная сеть глубиной 8 слоев. Вы можете загрузить предварительно обученную версию сети, обученную более чем миллиону изображений, из базы данных ImageNet.

Как функция активации используется Relu вместо арктангенса для добавления в модель нелинейности. За счет этого при одинаковой точности метода скорость становится в 6 раз быстрее.

Использование дропаута вместо регуляризации решает проблему переобучения. Однако время обучения удваивается с показателем дропаута 0,5.

Производится перекрытие объединений для уменьшения размера сети. За счет этого уровень ошибок первого и пятого уровней снижаются до 0,4% и 0,3%, соответственно.

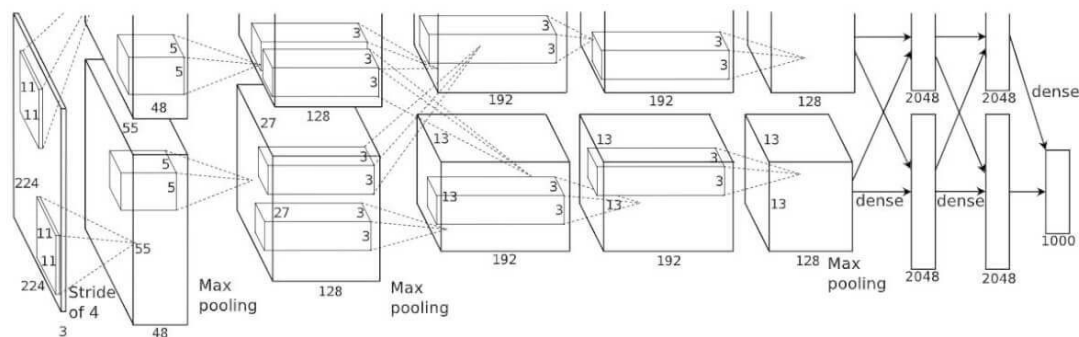


Рисунок 3.26 – Архитектура AlexNet

Архитектура сети приведена на рисунке 3.26 AlexNet содержит восемь слоев с весовыми коэффициентами. Первые пять из них сверточные, а остальные три — полносвязные. Выходные данные пропускаются через функцию потерь softmax, которая формирует распределение 1000 меток классов. Сеть максимизирует многолинейную логистическую регрессию, что эквивалентно максимизации среднего по всем обучающим случаям логарифма вероятности правильной маркировки по распределению ожидания. Ядра второго, четвертого и пятого сверточных слоев связаны только с теми картами ядра в предыдущем слое, которые находятся на одном и том же графическом процессоре. Ядра третьего сверточного слоя связаны со всеми картами ядер второго слоя. Нейроны в полносвязных слоях связаны со всеми нейронами предыдущего слоя.

Был выполнен анализ на производительность и энергопотребления в двух режимах: 15 Вт (HD) и MAX – IN (HD). На рисунках 3.27, 3.28 и 3.29 показаны результаты исследования при загрузке процессора партиями в режиме 15 Вт, состоящими из 1, 2, 4, 16, 32, 64 и 128 кадров изображения, т. е. видеопоток с разной частотой кадров.

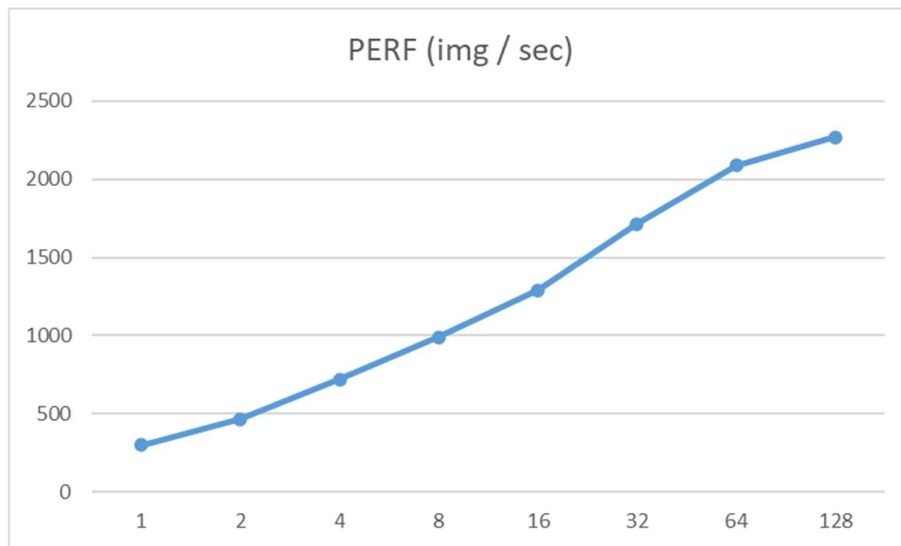


Рисунок 3.27 – График производительности изображения в секунду метода AlexNet на режиме 15 Вт

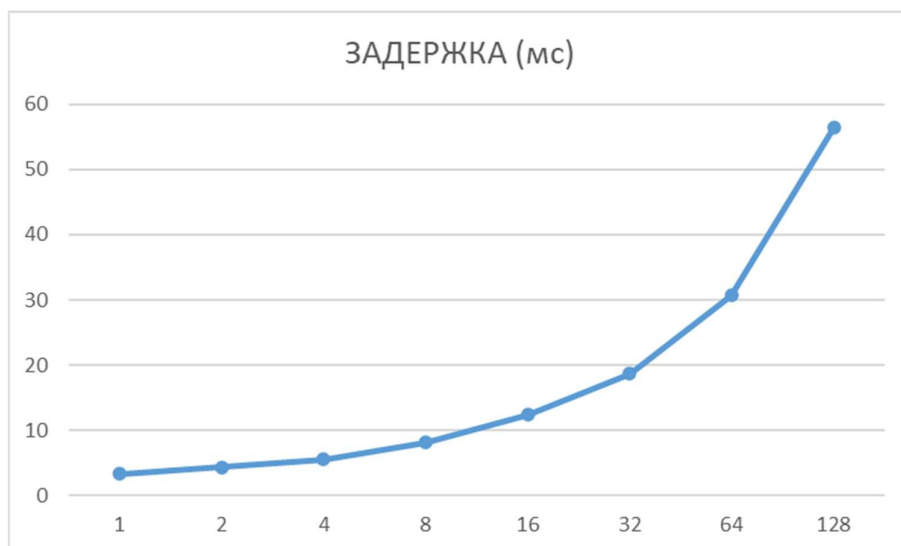


Рисунок 3.28 – График задержки выполнения партии потока метода AlexNet на режиме 15 Вт



Рисунок 3.29 – График энергопотребления метода AlexNet на режиме 15 Вт

На рисунках 3.30, 3.31 и 3.33 показаны результаты исследования при загрузке процессора партиями в режиме MAX - IN, состоящими также из 1, 2, 4, 16, 32, 64 и 128 кадров изображения.

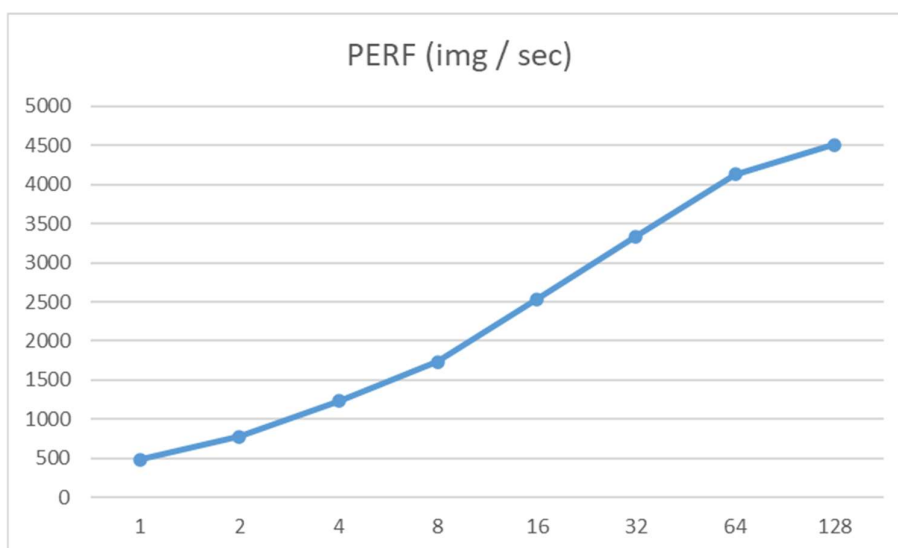


Рисунок 3.30 – График производительности изображения в секунду метода AlexNet на режиме MAX – IN

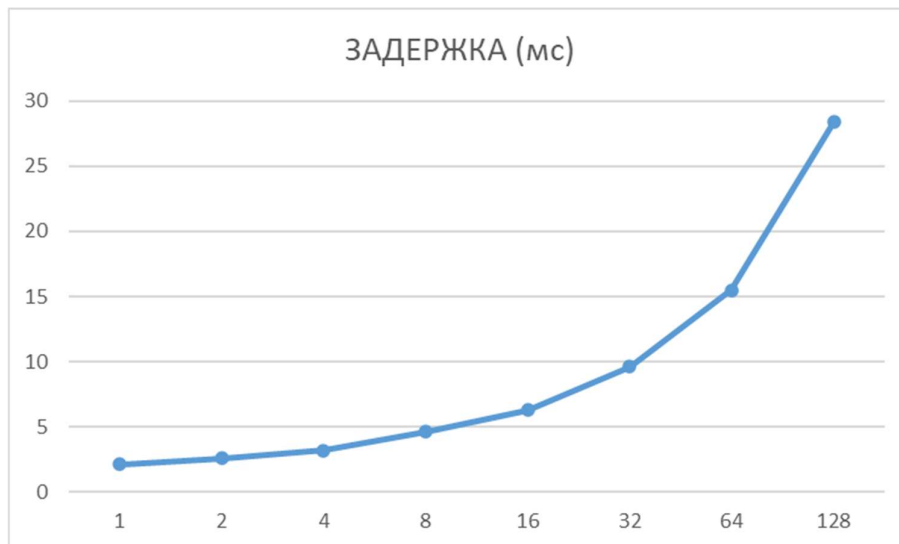


Рисунок 3.31 – График задержки выполнения партии потока метода AlexNet на режиме MAX – IN



Рисунок 3.32 – График энергопотребления метода AlexNet на режиме MAX – IN

3.2 Выбор оптимального метода для распознавания объектов

Исходя из полученных данных исследования загруженности процессора, были выявлены результаты электропотребления, производительности и задержки обработки видеопотока с разной частотой кадров имеющие разные размеры партии.

Рассмотренные методы и инструменты имеют разные ограничения и производительность.

Оптимальным методом решения распознавания объектов и его сегментации подходит ResNet – 18 FCN.

Во – первых, метод позволяет вводить изображение размером 2048*1024 и производить распознавание объекта его дальнейшим сегментацией.

Во – вторых, исходя из полученных данных исследования метод обладает низким потреблением электроэнергии, высокой производительной и минимальной задержкой на 15 Вт и MAX – IN режимах.

3.3 Распознавания объектов с использованием датасета

В качестве производимой работы по распознаванию объектов был применен сформированный датасет и обучена нейронная сеть для распознавания простых объектов с использованием RGB камеры.

В операционной системе Linux написан программный код на языке Python и загружен сформированный датасет. Был использован метод ResNet – 18 FCN имеющий 18 слоев весовыми коэффициентами. С помощью сформированным датасетом процесс обучения происходит быстрее, чем без него.



Рисунок 3.33 – Объект (пакет) на воде с отблесками солнца



Рисунок 3.34 – Объект (пакет) на суше

Фрагмент программного кода определения объекта приведен в ПРИЛОЖЕНИИ А.

ЗАКЛЮЧЕНИЕ

В ходе выполнения магистерской работы были выполнены все поставленные задачи. Разработана микропроцессорная система обнаружения разливов нефтепродуктов.

В процессе разработки обязательным условием стояло соблюдение норм отказоустойчивости и высокой производительностью при минимальном энергопотреблении. Были разработаны схемы, печатная плата и построена виртуальная система распознавания объектов разных форм.

Задачи, которые закладывались в магистерской диссертации были успешно выполнены и проверены на практике.

Можно сделать вывод, что разработанная система полностью соответствует все нормам и правилам. Работа всего программно-аппаратного обеспечения дало хороший результат в рамках данной работы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Сосонкин В. Л., Мартинов Г. М. Системы числового программного управления. Учебное пособие для вузов. Изд. Логос, 2005 г., 296 стр. ISBN 5-98704-012-4
2. <https://vt-metall.ru/articles/420-tochnost-plazmennoj-rezki-metalla>
3. ГОСТ 14792–80
- 4.

ПРИЛОЖЕНИЕ А

Листинг программы определения объектов методом ResNet – 18

```
"""Dilated
ResNet"""

import math
import torch
import torch.nn as nn
import torch.utils.model_zoo as model_zoo

__all__ = ['ResNet', 'resnet18', 'resnet34', 'resnet50', 'resnet101',
           'resnet152', 'BasicBlock', 'Bottleneck']

model_urls = {
    'resnet18': 'https://download.pytorch.org/models/resnet18-
5c106cde.pth',
    'resnet34': 'https://download.pytorch.org/models/resnet34-
333f7ec4.pth',
}

def conv3x3(in_planes, out_planes, stride=1):
    "3x3 convolution with padding"
    return nn.Conv2d(in_planes, out_planes, kernel_size=3,
stride=stride,
padding=1, bias=False)

class BasicBlock(nn.Module):
    """ResNet BasicBlock
    """
    expansion = 1
    def __init__(self, inplanes, planes, stride=1, dilation=1,
downsample=None, previous_dilation=1,
norm_layer=None):
        super(BasicBlock, self).__init__()
        self.conv1 = nn.Conv2d(inplanes, planes, kernel_size=3,
stride=stride,
padding=dilation, dilation=dilation,
bias=False)
        self.bn1 = norm_layer(planes)
        self.relu = nn.ReLU(inplace=True)
        self.conv2 = nn.Conv2d(planes, planes, kernel_size=3, stride=1,
padding=previous_dilation,
dilation=previous_dilation, bias=False)
        self.bn2 = norm_layer(planes)
```

```

        self.downsample = downsample
        self.stride = stride

    def forward(self, x):
        residual = x

        out = self.conv1(x)
        out = self.bn1(out)
        out = self.relu(out)

        out = self.conv2(out)
        out = self.bn2(out)

        if self.downsample is not None:
            residual = self.downsample(x)

        out += residual
        out = self.relu(out)

    return out

class Bottleneck(nn.Module):
    """ResNet Bottleneck
    """
    # pylint: disable=unused-argument
    expansion = 4
    def __init__(self, inplanes, planes, stride=1, dilation=1,
                 downsample=None, previous_dilation=1, norm_layer=None):
        super(Bottleneck, self).__init__()
        self.conv1 = nn.Conv2d(inplanes, planes, kernel_size=1,
                                bias=False)
        self.bn1 = norm_layer(planes)
        self.conv2 = nn.Conv2d(
            planes, planes, kernel_size=3, stride=stride,
            padding=dilation, dilation=dilation, bias=False)
        self.bn2 = norm_layer(planes)
        self.conv3 = nn.Conv2d(
            planes, planes * 4, kernel_size=1, bias=False)
        self.bn3 = norm_layer(planes * 4)
        self.relu = nn.ReLU(inplace=True)
        self.downsample = downsample
        self.dilation = dilation
        self.stride = stride

    def _sum_each(self, x, y):

```

```

    assert(len(x) == len(y))
    z = []
    for i in range(len(x)):
        z.append(x[i]+y[i])
    return z

def forward(self, x):
    residual = x

    out = self.conv1(x)
    out = self.bn1(out)
    out = self.relu(out)

    out = self.conv2(out)
    out = self.bn2(out)
    out = self.relu(out)

    out = self.conv3(out)
    out = self.bn3(out)

    if self.downsample is not None:
        residual = self.downsample(x)

    out += residual
    out = self.relu(out)

    return out

```

```

class ResNet(nn.Module):
    """Dilated Pre-trained ResNet Model, which produces the stride of 8
    featuremaps at conv5.
    Parameters
    -----
    block : Block
        Class for the residual block. Options are BasicBlockV1,
        BottleneckV1.
    layers : list of int
        Numbers of layers in each block
    classes : int, default 1000
        Number of classification classes.
    dilated : bool, default False
        Applying dilation strategy to pretrained ResNet yielding a
        stride-8 model,
        typically used in Semantic Segmentation.
    norm_layer : object

```

```

        Normalization layer used in backbone network (default:
:class:`mxnet.gluon.nn.BatchNorm`;
        for Synchronized Cross-GPU BatchNormalization).
Reference:
    - He, Kaiming, et al. "Deep residual learning for image
recognition." Proceedings of the IEEE conference on computer vision and
pattern recognition. 2016.
    - Yu, Fisher, and Vladlen Koltun. "Multi-scale context
aggregation by dilated convolutions."
    """
    # pylint: disable=unused-variable
    def __init__(self, block, layers, num_classes=1000, dilated=True,
                  deep_base=True, norm_layer=nn.BatchNorm2d,
output_size=8):
        self.inplanes = 128 if deep_base else 64
        super(ResNet, self).__init__()
        if deep_base:
            self.conv1 = nn.Sequential(
                nn.Conv2d(3, 64, kernel_size=3, stride=2, padding=1,
bias=False),
                norm_layer(64),
                nn.ReLU(inplace=True),
                nn.Conv2d(64, 64, kernel_size=3, stride=1, padding=1,
bias=False),
                norm_layer(64),
                nn.ReLU(inplace=True),
                nn.Conv2d(64, 128, kernel_size=3, stride=1, padding=1,
bias=False),
            )
        else:
            self.conv1 = nn.Conv2d(3, 64, kernel_size=7, stride=2,
padding=3,
                                bias=False)
        self.bn1 = norm_layer(self.inplanes)
        self.relu = nn.ReLU(inplace=True)
        self.maxpool = nn.MaxPool2d(kernel_size=3, stride=2, padding=1)
        self.layer1 = self._make_layer(block, 64, layers[0],
norm_layer=norm_layer)
        self.layer2 = self._make_layer(block, 128, layers[1], stride=2,
norm_layer=norm_layer)

        dilation_rate = 2
        if dilated and output_size <= 8:
            self.layer3 = self._make_layer(block, 256, layers[2],
stride=1,

```



```

dilation=dilation_rate,
norm_layer=norm_layer)
    dilation_rate *= 2
else:
    self.layer3 = self._make_layer(block, 256, layers[2],
stride=2,
norm_layer=norm_layer)

    if dilated and output_size <= 16:
        self.layer4 = self._make_layer(block, 512, layers[3],
stride=1,
dilation=dilation_rate,
norm_layer=norm_layer)
    else:
        self.layer4 = self._make_layer(block, 512, layers[3],
stride=2,
norm_layer=norm_layer)

self.avgpool = nn.AvgPool2d(7, stride=1)
self.fc = nn.Linear(512 * block.expansion, num_classes)

for m in self.modules():
    if isinstance(m, nn.Conv2d):
        n = m.kernel_size[0] * m.kernel_size[1] * m.out_channels
        m.weight.data.normal_(0, math.sqrt(2. / n))
    elif isinstance(m, norm_layer):
        m.weight.data.fill_(1)
        m.bias.data.zero_()

def _make_layer(self, block, planes, blocks, stride=1, dilation=1,
norm_layer=None):
    downsample = None
    if stride != 1 or self.inplanes != planes * block.expansion:
        downsample = nn.Sequential(
            nn.Conv2d(self.inplanes, planes * block.expansion,
kernel_size=1, stride=stride, bias=False),
            norm_layer(planes * block.expansion),
        )

    layers = []
    if dilation == 1 or dilation == 2:
        layers.append(block(self.inplanes, planes, stride,
dilation=1,
downsample=downsample,
previous_dilation=dilation, norm_layer=norm_layer))
    elif dilation == 4:

```

```

        layers.append(block(self.inplanes, planes, stride,
dilation=2,
                            downsample=downsample,
previous_dilation=dilation, norm_layer=norm_layer))
    else:
        raise RuntimeError("=> unknown dilation size:
{}".format(dilation))

    self.inplanes = planes * block.expansion
    for i in range(1, blocks):
        layers.append(block(self.inplanes, planes,
dilation=dilation, previous_dilation=dilation,
                            norm_layer=norm_layer))

    return nn.Sequential(*layers)

def forward(self, x):
    x = self.conv1(x)
    x = self.bn1(x)
    x = self.relu(x)
    x = self.maxpool(x)

    x = self.layer1(x)
    x = self.layer2(x)
    x = self.layer3(x)
    x = self.layer4(x)

    x = self.avgpool(x)
    x = x.view(x.size(0), -1)
    x = self.fc(x)

    return x

def resnet18(pretrained=False, **kwargs):
    """Constructs a ResNet-18 model.
    Args:
        pretrained (bool): If True, returns a model pre-trained on
ImageNet
    """
    model = ResNet(BasicBlock, [2, 2, 2, 2], **kwargs)
    if pretrained:
        model.load_state_dict(model_zoo.load_url(model_urls['resnet18']))
    return model

```

```

def resnet34(pretrained=False, **kwargs):
    """Constructs a ResNet-34 model.
    Args:
        pretrained (bool): If True, returns a model pre-trained on
ImageNet
    """
    model = ResNet(BasicBlock, [3, 4, 6, 3], **kwargs)
    if pretrained:
        model.load_state_dict(model_zoo.load_url(model_urls['resnet34']))
    return model

def resnet50(pretrained=False, root='~/encoding/models', **kwargs):
    """Constructs a ResNet-50 model.
    Args:
        pretrained (bool): If True, returns a model pre-trained on
ImageNet
    """
    model = ResNet(Bottleneck, [3, 4, 6, 3], **kwargs)
    if pretrained:
        from ..models.model_store import get_model_file
        model.load_state_dict(torch.load(
            get_model_file('resnet50', root=root)), strict=False)
    return model

def resnet101(pretrained=False, root='~/encoding/models', **kwargs):
    """Constructs a ResNet-101 model.
    Args:
        pretrained (bool): If True, returns a model pre-trained on
ImageNet
    """
    model = ResNet(Bottleneck, [3, 4, 23, 3], **kwargs)
    if pretrained:
        from ..models.model_store import get_model_file
        model.load_state_dict(torch.load(
            get_model_file('resnet101', root=root)), strict=False)
    return model

def resnet152(pretrained=False, root='~/encoding/models', **kwargs):
    """Constructs a ResNet-152 model.
    Args:

```

```
pretrained (bool): If True, returns a model pre-trained on
ImageNet
"""
model = ResNet(Bottleneck, [3, 8, 36, 3], **kwargs)
if pretrained:
    from ..models.model_store import get_model_file
    model.load_state_dict(torch.load(
        get_model_file('resnet152', root=root)), strict=False)
return model
```

ПРИЛОЖЕНИЕ Б

**Принципиальная электрическая схема микропроцессорной системы
обнаружения разливов нефтепродуктов**

ПРИЛОЖЕНИЕ В

Печатная плата низкого уровня обработки метаданных