

**RoboCup Junior Soccer Light Weight**

**Team Description Paper**

**Команда**

**Asterisk\***

**Состав:**

Балакирский Аркадий - Программист.

Мухачев Денис – Капитан, инженер, электронщик.

**Руководители:**

Романько Павел Николаевич

Устинов Илья Дмитриевич

Викторов Борис Викторович

**Организация:**

“Президентский” физико-математический лицей №239

Санкт-Петербург, Россия

**Введение**

Команда Asterisk\* выступает на соревнованиях RoboCup Junior Soccer Lightweight от организации “Президентский” физико-математический лицей №239. Команда уже принимала участие в Открытых состязаниях Санкт-Петербурга по робототехнике 2024 и в XI Минском Открытом роботурнире. Подробное описание каждого участника команды:

1) Балакирский Аркадий Владимирович – Программирование основного микроконтроллера, программирование камеры.

Telegram: @b\_asterisk

Почта: [integralplusc@gmail.com](mailto:integralplusc@gmail.com)

2) Мухачев Денис Павлович – Создание конструкции, пайка и разработка электроники.

Telegram: @foursimb

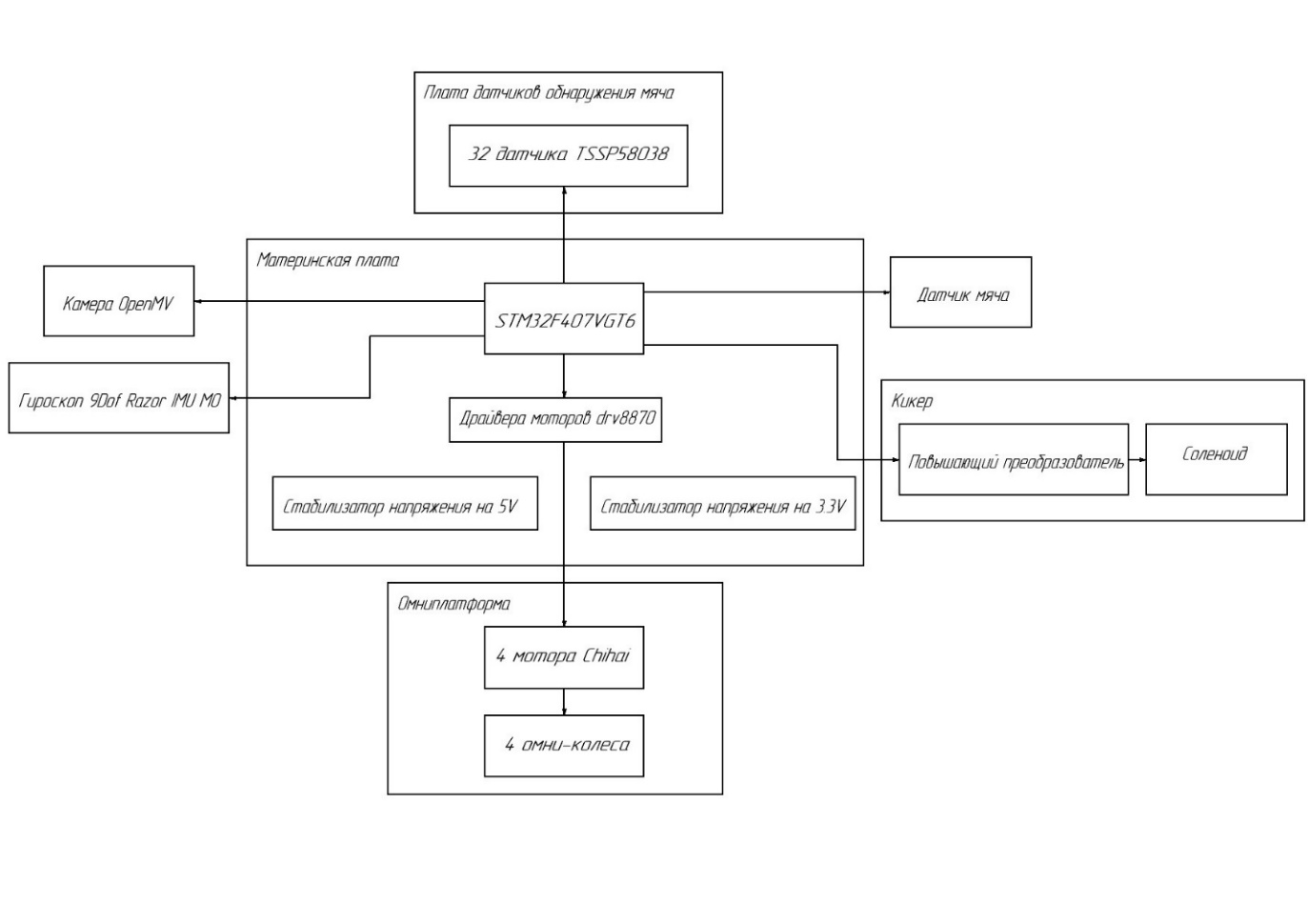
Почта: [mookhachev.denis@gmail.com](mailto:Mookhachev.denis@gmail.com)

Общее фото команды:



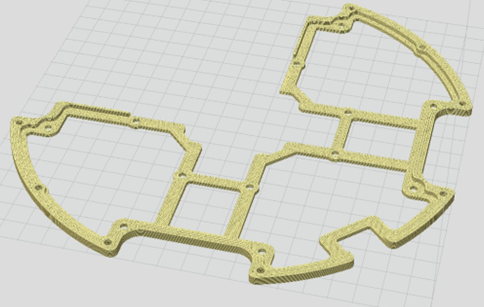
**Конструкция**

Ниже приведена структурная схема робота:



Для проектирования был выбран САПР Solid Works.

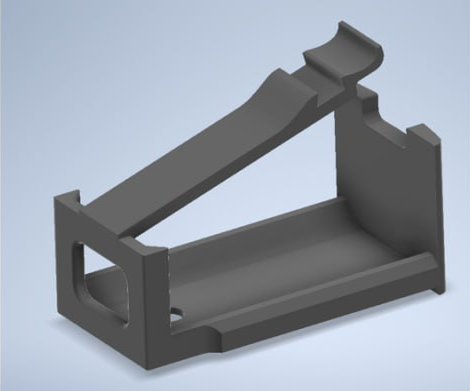
На роботе закреплены 4 перпендикулярно расположенных мотора. Для обеспечения жесткости конструкции сверху и снизу они прикреплены к алюминиевым уровням жесткости, которые выполняют роль каркаса. Для защиты материнской платы от повреждений извне используется специальная боковая защита, сделанная из пластика, и напечатанная на 3D-принтере. Ниже дано изображение этой боковой защиты:



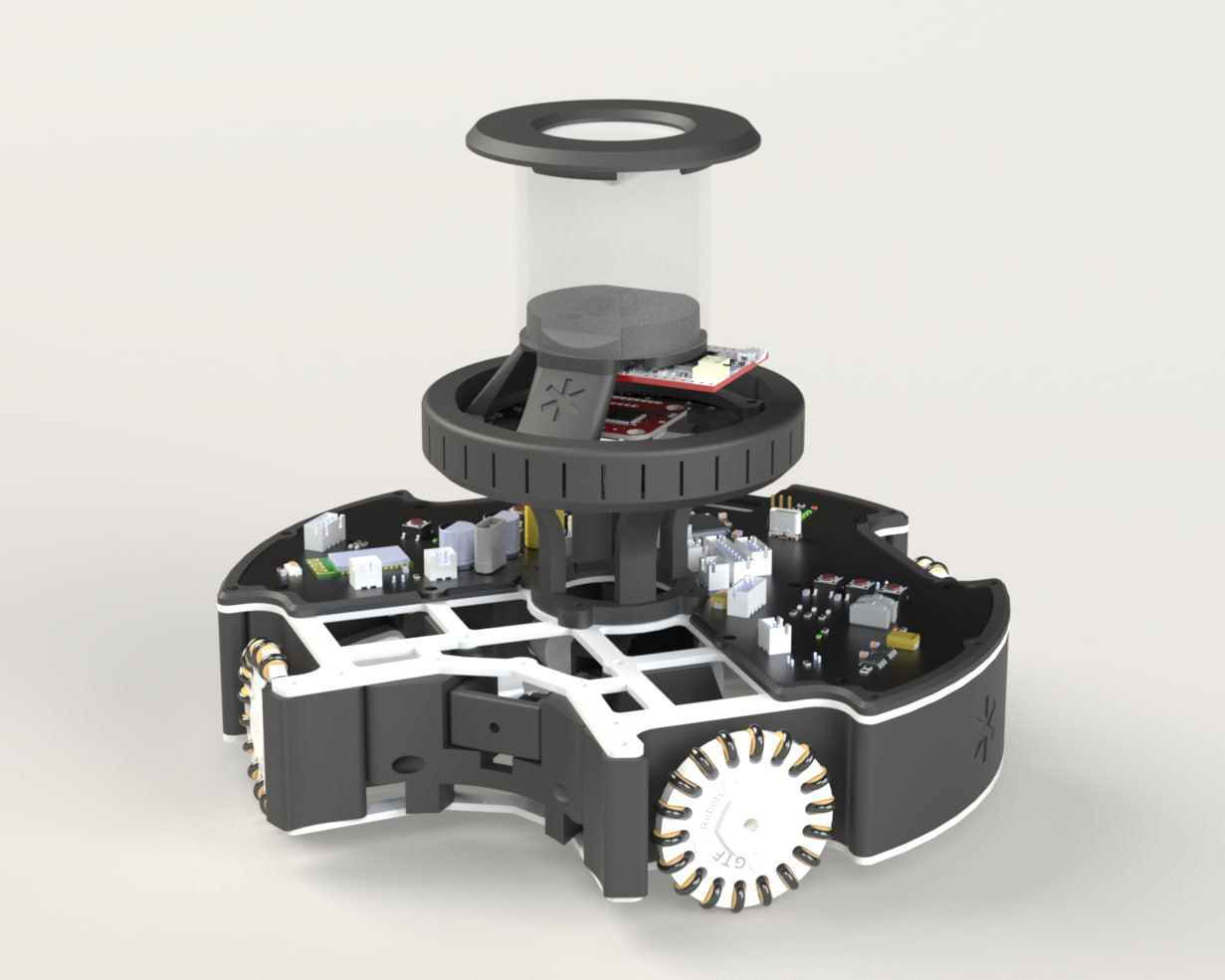
Также важной частью робота является его круговая защита, расположенная по периметру робота между алюминиевыми уровнями, она обеспечивает сохранность внутренних деталей робота. Для обеспечения свободного перемещения робота по соревновательному полигону было принято решение использовать омни-колеса. Ниже изображена омни-платформа без боковой защиты и колес:



Также в задней части между уровнями жесткости есть крепление для аккумулятора с направляющими и защелкивающимся механизмом для фиксации аккумулятора внутри робота. Это крепление позволяет быстро и удобно вставлять и доставать аккумулятор из робота так, чтобы аккумулятор не мог выпасть во время матча. Ниже приведено изображение крепления:



Над верхним алюминиевым уровнем закреплена материнская плата. Также над материнской платой расположена плата датчиков мяча, где расположено 32 датчика TSSP58038, образующих круг. Каждый датчик находится в дополнительном пластиковом корпусе с щелью спереди, чтобы датчик обнаруживал мяч, только тогда, когда мяч находится перед датчиком. На уровне с платой датчиков мяча расположен датчик-гироскоп 9Dof Razor IMU M0. Ниже приведена изображение робота из Inventor’а:



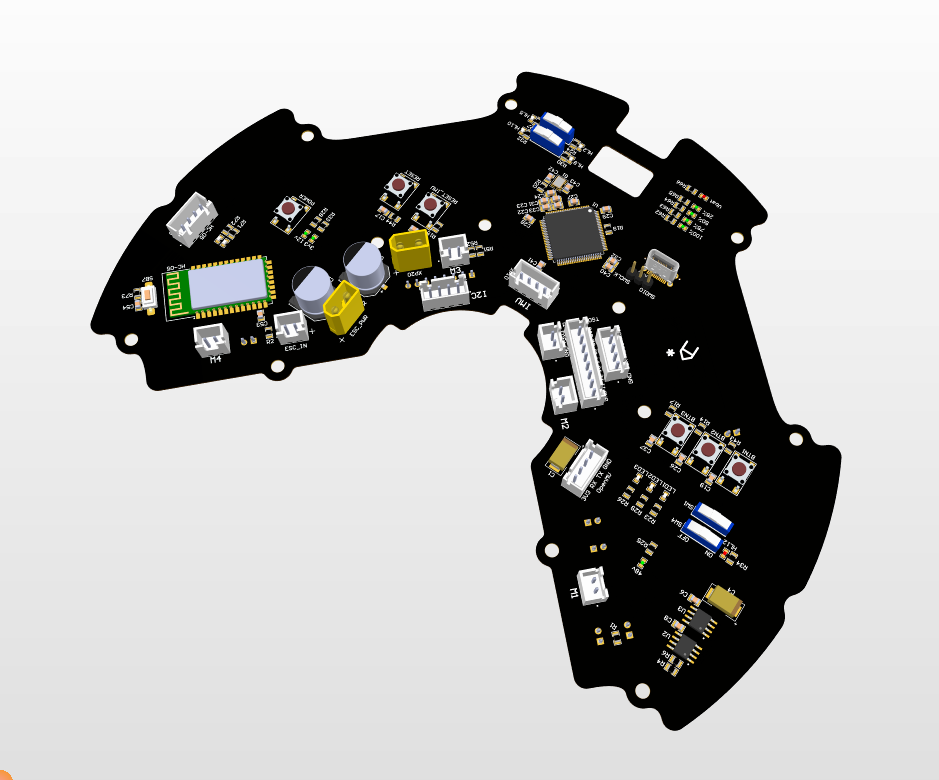
Также посередине расположена камера, ее крепление и конусовидное зеркало, расположенное ровно над камерой с помощью прозрачной трубы. Зеркало крепится к трубе с помощью специального крепления, еще одной функцией которого является защита камеры робота от засветок.

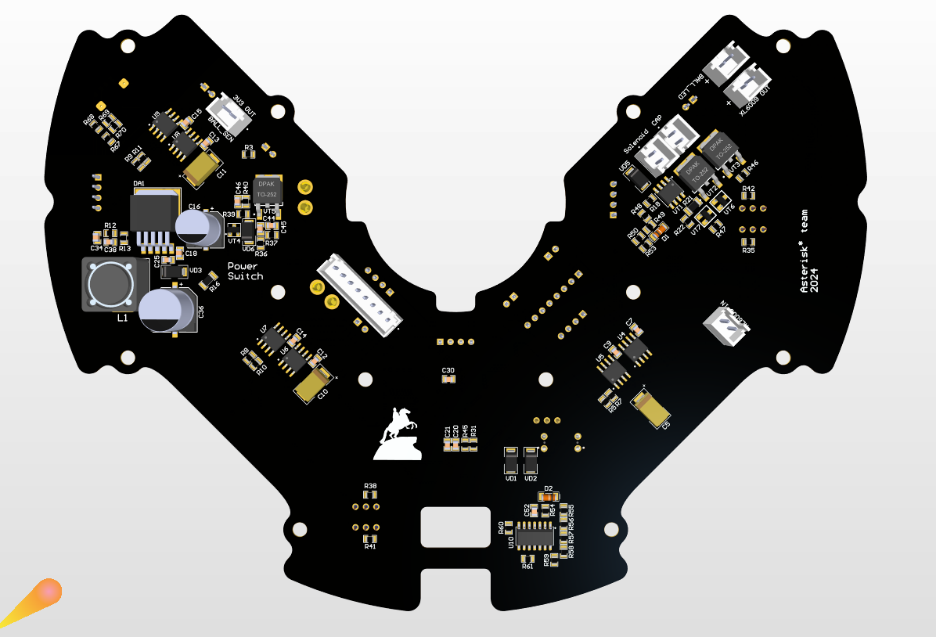
Еще одной из важных частей робота является его кикер. Он функционирует благодаря соленоиду, повышающему преобразователю и дополнительному конденсатору.

**Проектирование печатной платы**

Для проектирования плат был выбран САПР Altium Designer.

На основной плате располагается микроконтроллер STM32F407VGT6, к нему подключены все датчики. Также, на плате расположено 8 драйверов моторов drv8870, по 2 драйвера на мотор. На плате расположено 2 стабилизатора напряжения на 3.3 вольта и 5 вольт.





На плате имеются разъемы под USART, ADC, I2C и под обычные пины. Также на плате присутствуют программируемые светодиоды, программируемые кнопки и переключатели.

**Программирование**

**Программирование основного микроконтроллера робота STM32F407**

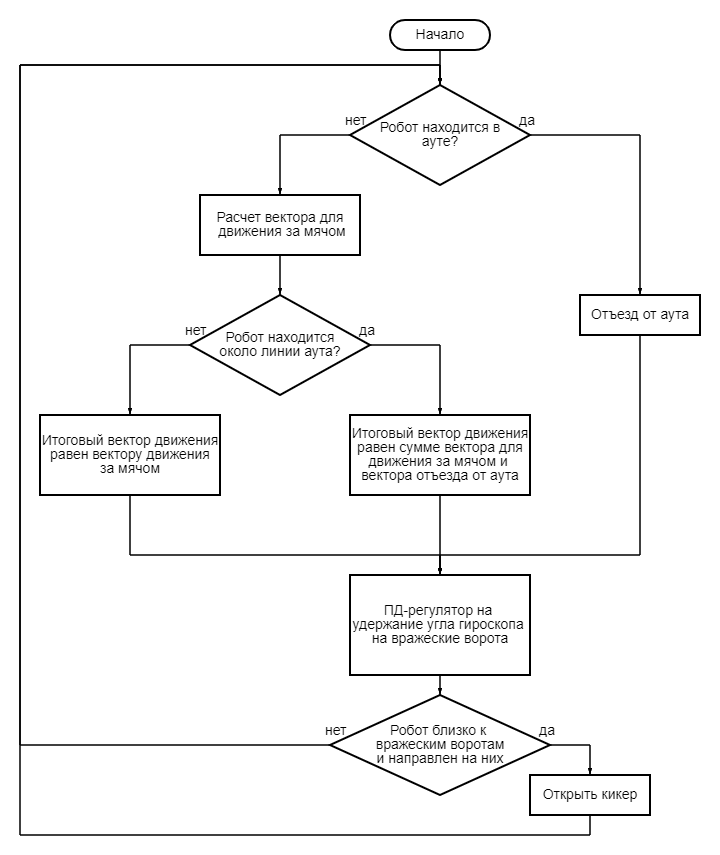
Программа основного микроконтроллера написана на языке C++ в среде Keil μVision5. Программа разбита на множество классов и именных пространств, что обеспечивает создание нового кода и читаемость. Например, есть классы для работы со всеми каналами передачи данных на роботе, класс работы с омни-платформой, кикером, камерой, гироскопом, векторами и так далее. Вся стратегия робота написана в именном пространстве Asterisk.

В начале программы происходит инициализация всех датчиков и глобальных переменных. Также первые 40 секунд работы робота происходит калибровка значения датчика-гироскопа 9Dof Razor IMU M0.

В самом начале цикла программы робот начинает получать значения со всех датчиков, в том числе происходит обработка значений, получаемых с камеры, для получения точных координат робота на поле относительно ворот.

**Программирование нападающего**

Ниже представлена блок-схема программы нападающего:

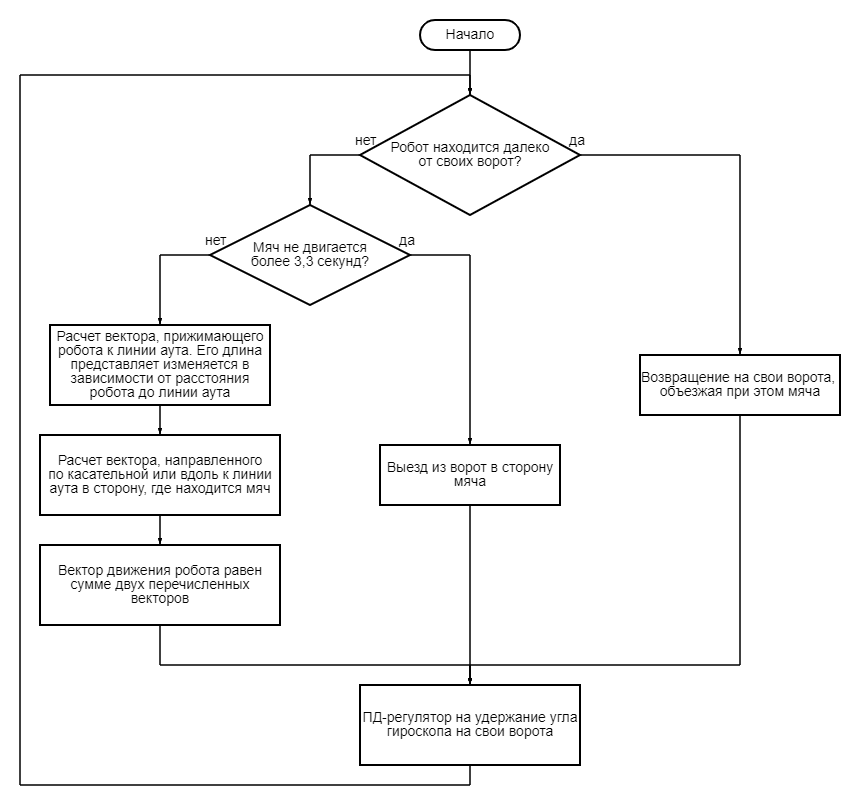


Зона аутов в программе определяется исключительно координатным методом, то есть заранее записаны ограничения координаты X по бокам, координаты Y сверху и снизу, а также определена зона вокруг своих и вражеских ворот. Каждая такая зона определена отрезком (прямой участок аута перед воротами) и двумя дугами окружности, центр которой находится в центре ворот. Во всех случаях вектор отъезда от аута перпендикулярен вектору линии аута.

Вектор движения за мячом высчитывается следующим образом: если мяч находится далеко от робота, то итоговый вектор равен вектору от робота до мяча, если мяч находится близко к роботу, то высчитывается регулятор с экспоненциально изменяющейся ошибкой по углу и до мяча и по расстоянию до мяча.

**Программирование вратаря**

Ниже представлена блок-схема программы вратаря:



Основной целью вратаря является защита ворот, он старается заслонить собой ворота от мяча, для этого он проводит прямую от центра ворот к центру робота и проверяет угол отклонения мяча от этой прямой. С помощью этого угла составляется ошибка и подается на ПД-регулятор.

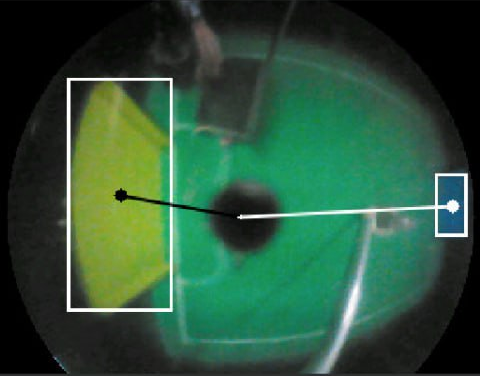
Еще на роботе установлена защита от попадания в нижний аут около своих ворот (эта защита сделана ограничением Y-координаты робота).

Также если вратарь находится очень далеко от ворот (например, в случае, когда после его попадания в аут его выставили на поле), он начинает двигаться к центру своих ворот.

**Программирование камеры**

Для получения координат робота на поле была выбрана камера OpenMV H7, программируемая на языке MicroPython в среде OpenMV IDE. Камера находит компоненты связности, подходящие под заданные цветовые параметры (то есть для желтых и синих ворот), вычисляет положение центров ворот на изображении. Далее камера отправляет на основной микроконтроллер расстояние и угол до каждого из ворот по каналу связи UART. Каждый отправленный пакет данных с камеры состоит из байта старта (числа 255), четырех чисел (2 расстояния и 2 угла), а также контрольная сумма (CRC8) из этих четырех чисел.

В основном микроконтроллере эти значения переводятся в декартовы координаты. Ниже изображено, как камера видит соревновательный полигон.



**Благодарности**

Команда Asterisk\* выражает благодарность нашим руководителям и спонсорам ООО “НПО “СтарЛайн”, Благотворительному фонду “Финист” и ПАО “Газпромнефть”.

