Глава 1. Концепция

Глава 2. Конструкция (с чертежами и моделями)

Глава 3. Электроника

3.1 Система питания

Для автономного передвижения и питания всех компонентов транспортного средства используется аккумулятор с техническими характеристиками, приведенными в таблице ниже.

Таблица 1. Технические характеристики аккумулятора.

|  |  |
| --- | --- |
| Наименование | Значение |
| Тип | LiPo |
| Внутренняя организация | 4S |
| Выходное напряжение | 16,8 В |
| Ток | 30С |
| Ёмкость | 5200 мАч |
| Тип разъема | XT60 |
| Наличие балансового разъема | Есть |

Получение необходимого стабилизированного напряжение 5 и 12 вольтовых линий выполнено с использованием стабилизаторов напряжения LM7805 и LM7812.

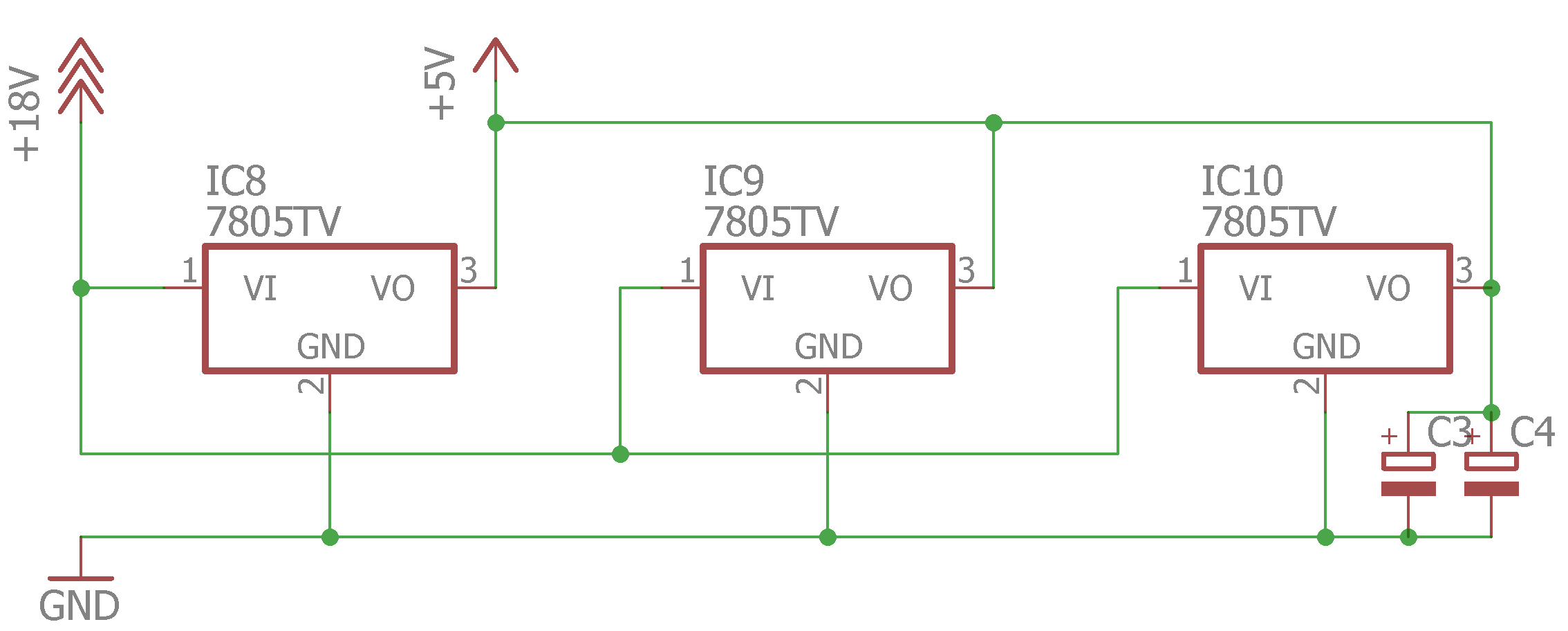


Рисунок 1. Электрическая схема системы питания 5В.

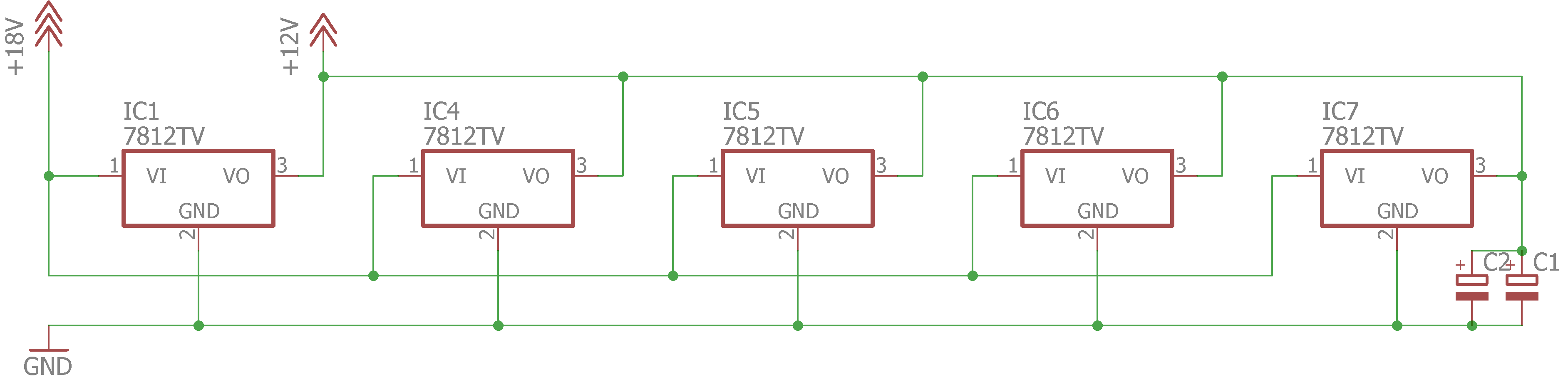


Рисунок 1. Электрическая схема системы питания 12В.

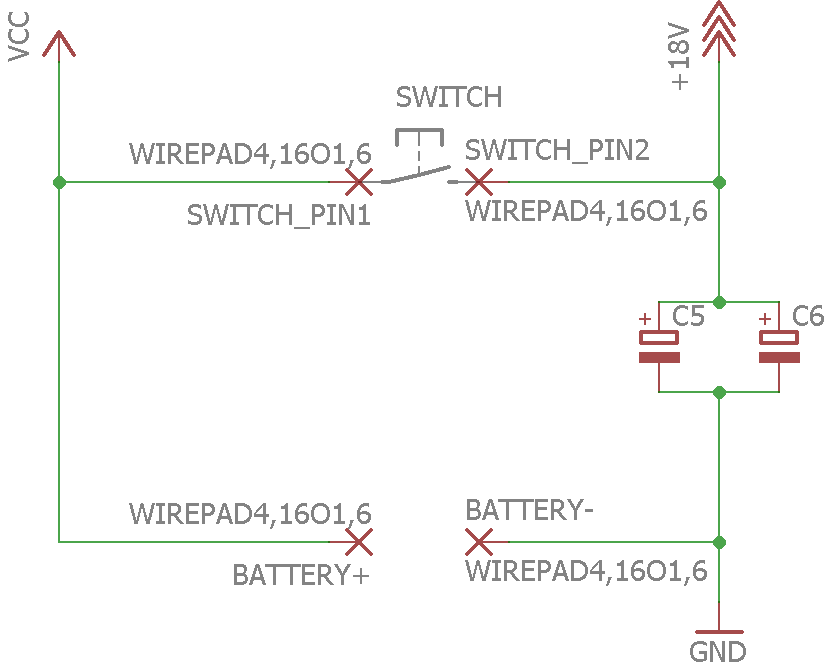


Рисунок 1. Электрическая схема подключения аккумулятора и выключателя системы питания.

3.2 Система управления низкого уровня (драйвер шасси)

Контроллером управления является Arduino STM32F103C8T6. Плата имеет выводы для подключения следующих устройств: ультразвуковые датчики, датчики числа оборотов колес, датчики направления движения, сервомотор рулевого управления, световая индикация, датчик заряда аккумулятора, драйвер управления электромоторами постоянного тока, подключение питания аккумулятора и место для подключения кнопки включения питания. Ниже на рисунке приведено обозначение выводов платы.

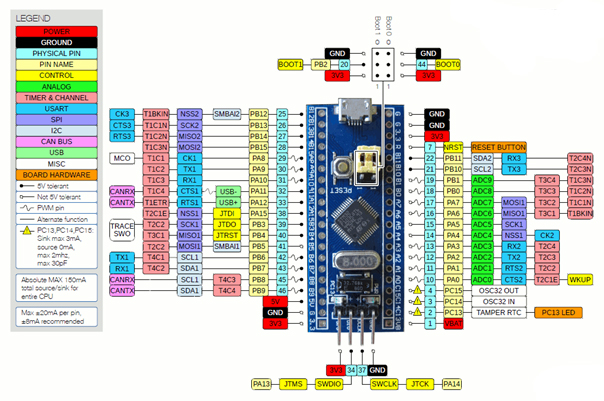


Рисунок 1. Обозначение выводов контроллера STM32F103C8T6.

Электрическая схема контроллера приведена на рисунке ниже.

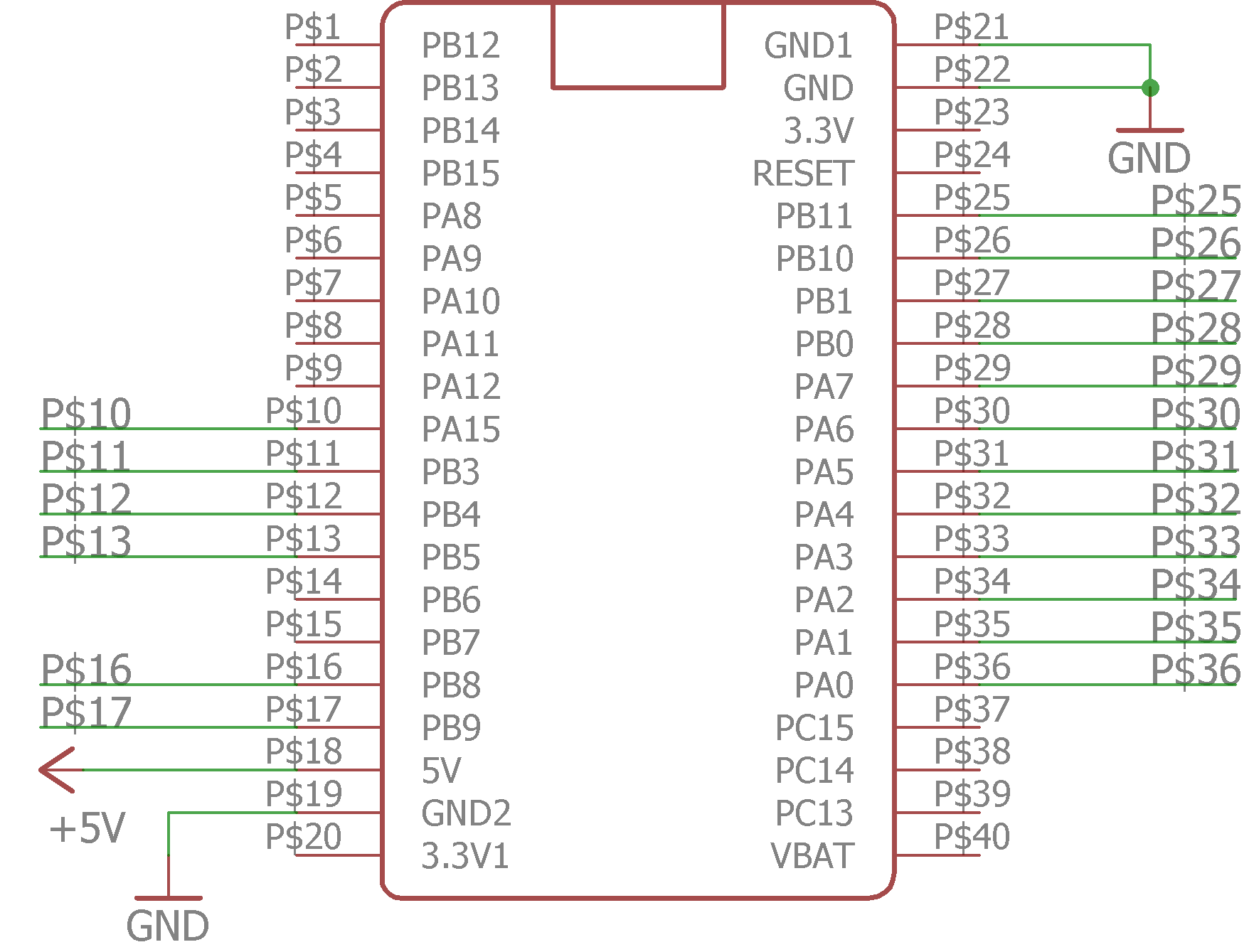


Рисунок 1. Электрическая схема контроллера.

Печатная плата имеет крепежные отверстия для крепления на шасси ТС. На рисунке ниже приведена печатная плата драйвера шасси.

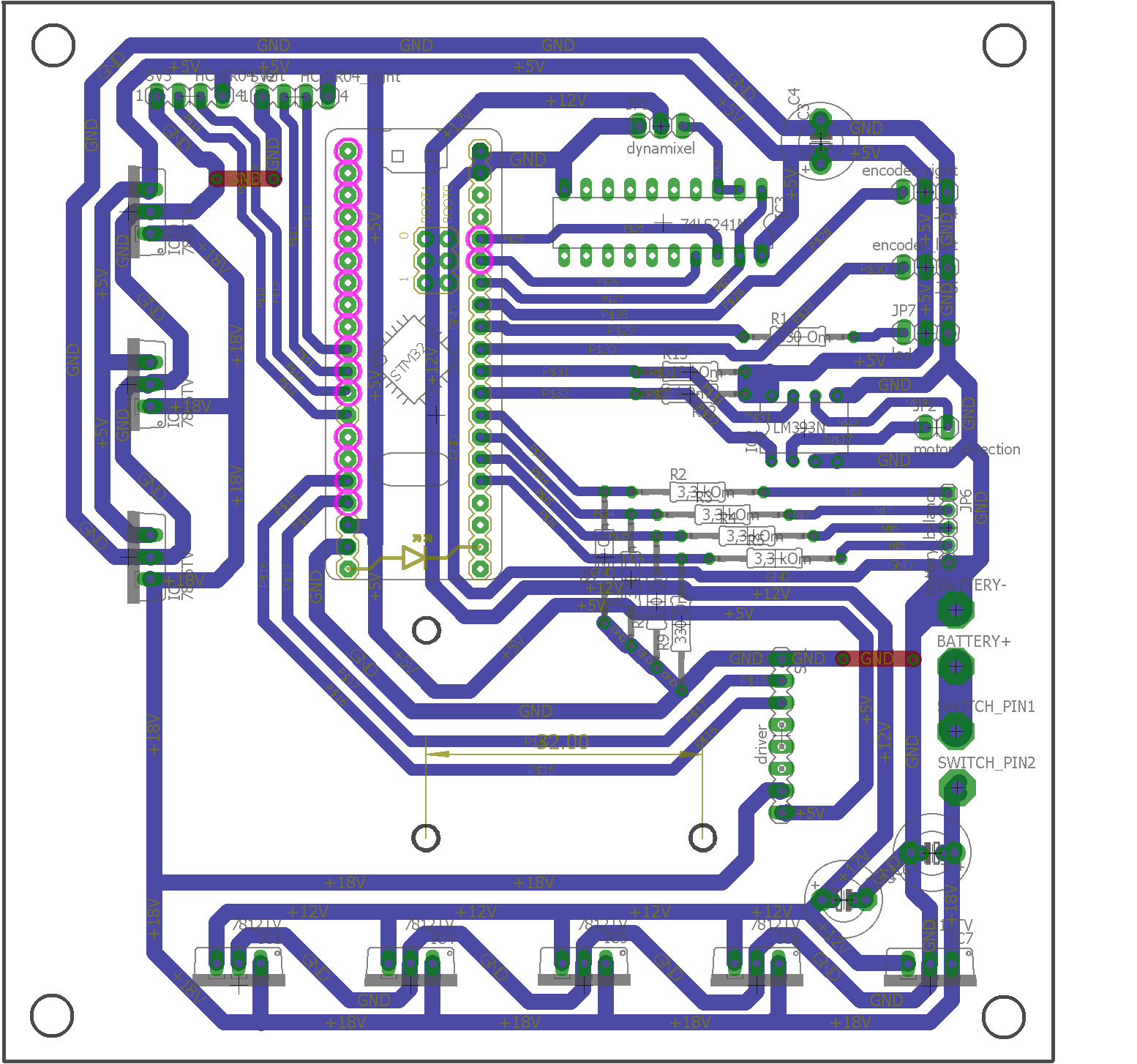


Рисунок 1. Печатная плата драйвера шасси.

3.2.1 Оптический датчик числа оборотов колеса

Для определения количества оборотов колеса используются бесконтактные оптические датчики положения. Они состоят из оптического излучателя и фотоприемника. Световой поток от излучателя попадает на фотоприемник, что вызывает определенное состояние датчика. Наличие непрозрачного объекта на пути светового луча приводит к изменению светового потока на фотоприемнике, а значит и к другому состоянию датчика.

Непрозрачным объектом является смоделированное образцовый диск с отверстиями, устанавливаемый на вал мотора (во внутренней части колеса).

Контроллер реагирует на прерывания при изменении состояния фотоприёмника с (HIGH на LOW, LOW на HIGH), что позволяет повысить точность в два раза.

Для щелевого оптрона и необходимой для него обвязки изготовлена печатная плата. Печатная плата приведена на рисунке ниже.

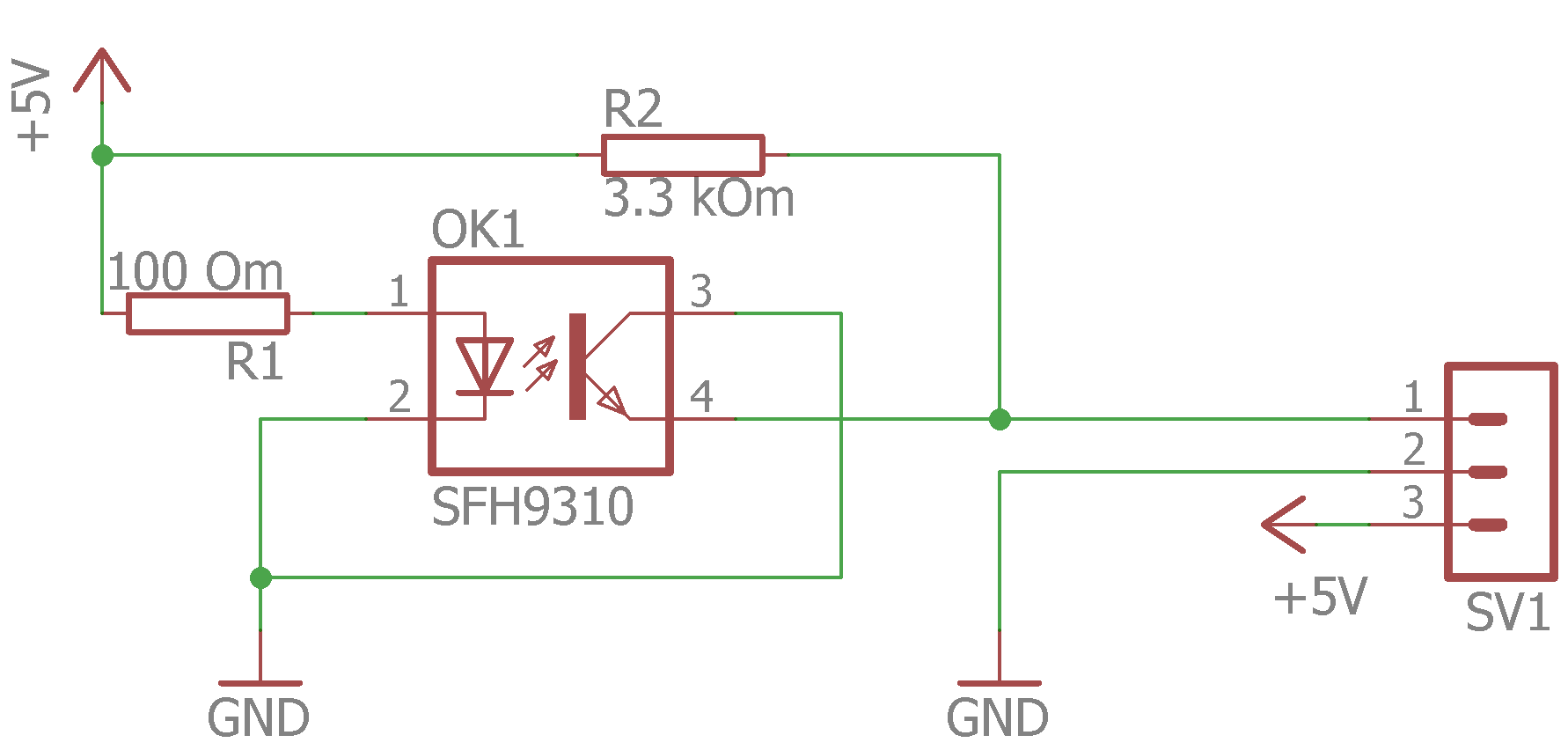


Рисунок 1. Электрическая схема оптического датчика числа оборотов.

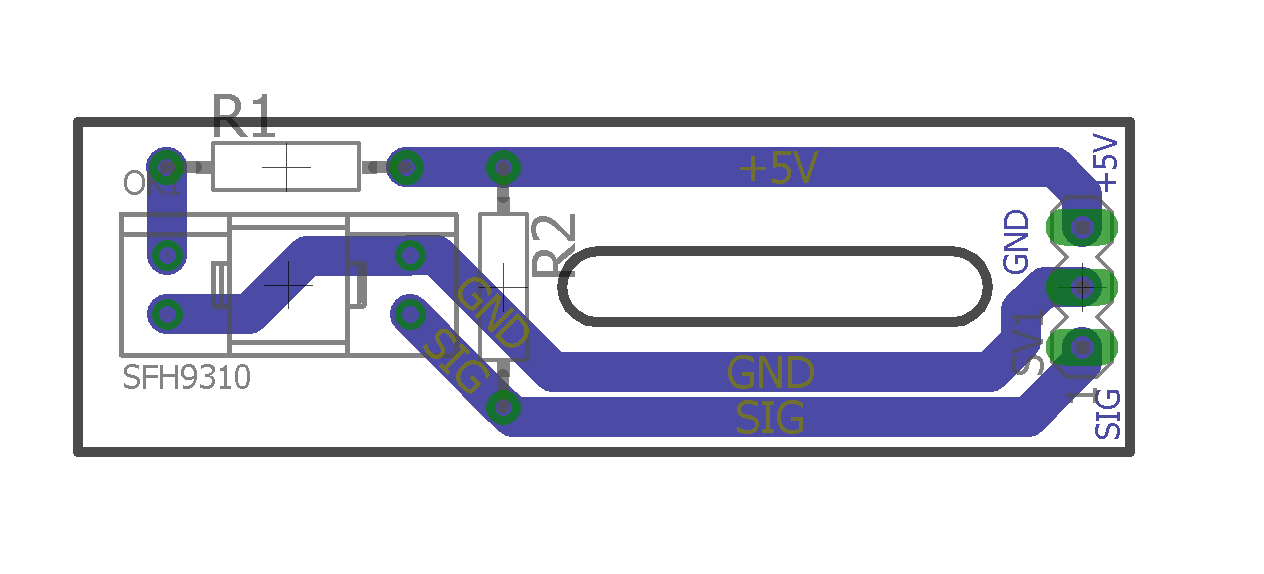


Рисунок 1. Печатная плата оптического датчика числа оборотов колеса.

Таблица 1. Описание выводов оптического датчика числа оборотов колеса.

|  |  |
| --- | --- |
| Обозначение | Наименование |
| +5V | положительный контакт питания 5В |
| GND | отрицательный контакт питания |
| SIG | выход с датчика |

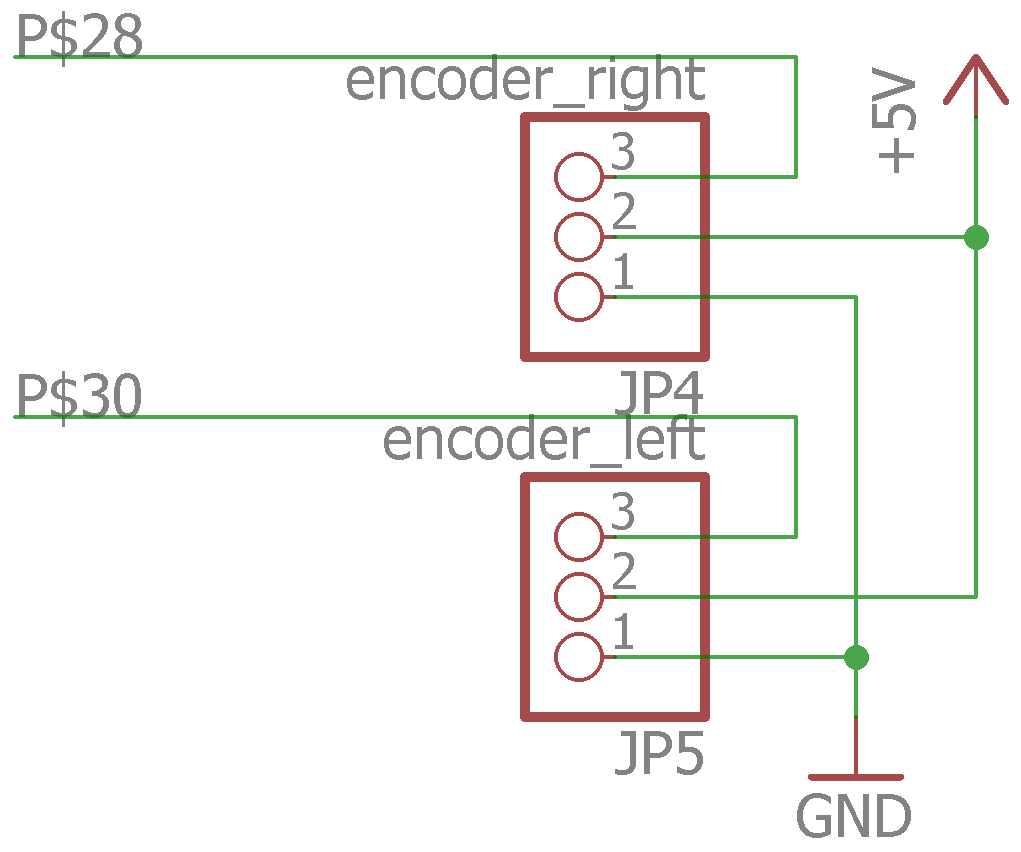


Рисунок 1. Схема подключения датчика числа оборотов колес к контроллеру.

* + 1. Драйвер управления двигателями постоянного тока

Для управления электродвигателями постоянного тока используется драйвер L298N. Технические характеристики приведены в таблице ниже.

Таблица 1. Технические характеристики драйвера

|  |  |
| --- | --- |
| Наименование | Значение |
| Размер платы | 30 мм x 54 мм |
| Питание платы | 5 В |
| Питание двигателей | 2.5 – 46 В |
| Рабочий ток на канал | до 2А |
| Логический "0" управляющего напряжения | 0...1.5 В |
| Логическая "1" управляющего напряжения | 2,3...7 В |
| Максимальная частота управляющего ШИМ | до 5 кГц |
| Защита от перегрева | Есть |

L298N содержит сразу два драйвера для управления электродвигателями (четыре независимых канала, объединенных в две пары). Имеет две пары входов для управляющих сигналов и две пары выходов для подключения электромоторов. Кроме того, у L298N есть два входа для включения каждого из драйверов. Эти входы используются для управления скоростью вращения электромоторов с помощью широтно-импульсной модуляции сигнала (ШИМ).

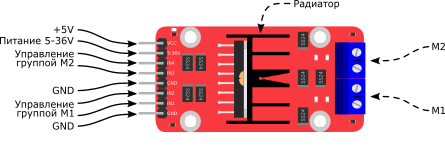


Рисунок 1. Драйвер управления электродвигателями постоянного тока L298N.

Таблица 1. Описание выводов драйвера управления электродвигателями постоянного тока

|  |  |
| --- | --- |
| Обозначение | Наименование |
| VCC | положительный контакт питания платы (управляющей логики) 5В |
| 5-36V | положительный контакт питания электродвигателей 5 – 36 В |
| GND | отрицательный контакт питания |
| IN1, IN2, IN3, IN4 | управление направлением вращения и скоростью двигателей |

L298N обеспечивает разделение электропитания для контроллера и для управляемых им двигателей, что позволяет подключить электродвигатели с большим напряжением питания, чем у контроллера. Разделение электропитания микросхем и электродвигателей может быть также необходимо для уменьшения помех, вызванных бросками напряжения, связанными с работой моторов.

Одна микросхема L298N способна управлять двумя двигателями по 2А каждый двигатель, а если задействовать параллельное включение для одного двигателя, то можно поднять максимальный ток до 4А.

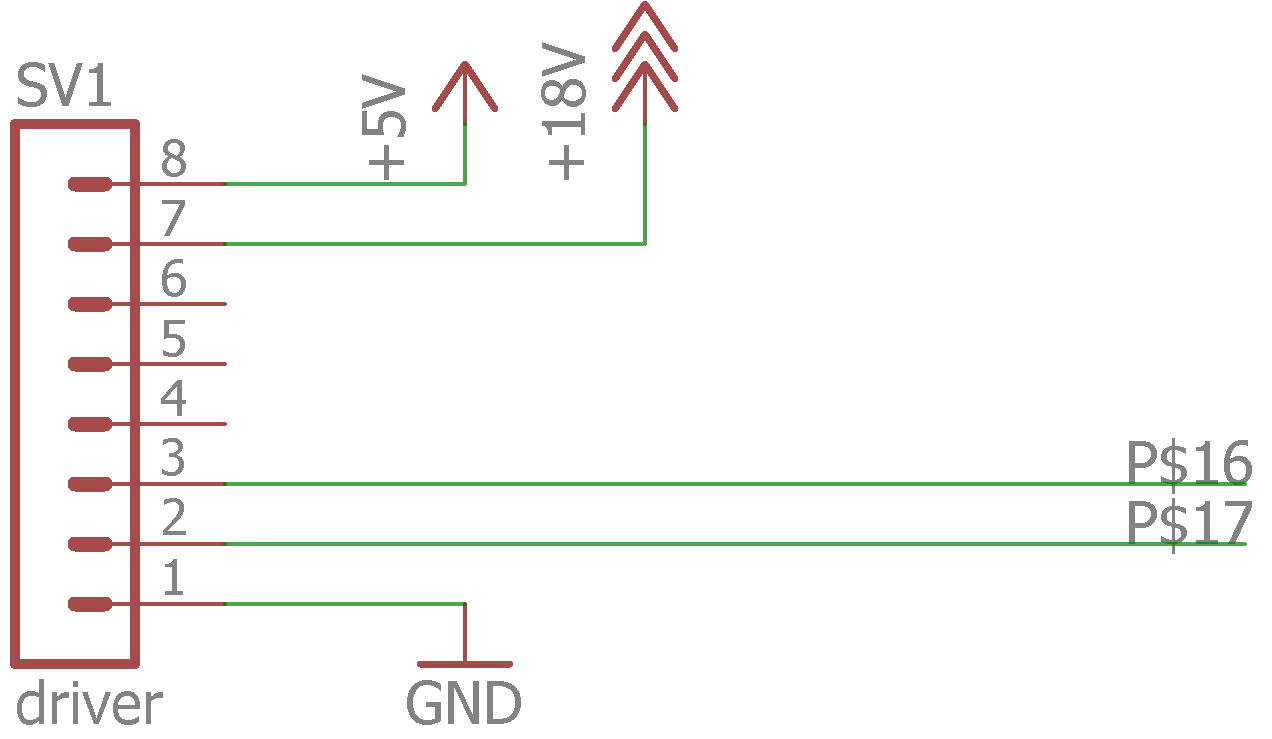


Рисунок 1. Схема подключения драйвера управления электродвигателями постоянного тока к контроллеру.

* + 1. Рулевое управление

Для рулевого управления используется сервомотор dynamixel AX-12a.

Таблица 1. Основные характеристики dynamixel AX-12a.

|  |  |
| --- | --- |
| Наименование | Значение |
| Рабочее напряжение | 9-12 В (рекомендуемое 11,1 В) |
| Передаточное число | 254:1 |
| Момент удержания | 1,5 Н\*м (12 В) |
| Ток удержания | 1,5 А |
| Скорость без нагрузки | 59 об/мин (12 В) |
| Минимальный угол поворота | 0,29 градуса х 1,024 |
| Угол поворота | в режиме актуатора - до 300 градусов, в режиме двигателя - без ограничений |
| Максимальный ток | 900 мА |
| Ток покоя | 50 мА |
| Управление | цифровое |
| Протокол | полудуплексный асинхронный последовательный (8N1) |
| Скорость интерфейса | 1Мбод/с |
| ID | 0...253 (по умолчанию ID = 1) |
| Обратная связь | положение, температура, нагрузка, напряжение и т.п. |
| Датчик положения | потенциометр |
| Вес | 54,6 г |
| Размеры | 32 мм х 50 мм х 40 мм |

Контроллером управления сервомотором является Arduino STM32F103C8T6. Промежуточным звеном между сервомотором и контроллером является цифровая микросхема серии ТТЛ (К555АП4). Микросхемы К555АП4 представляют собой два четырехканальных формирователя с тремя состояниями на выходе. Содержат 232 интегральных элемента. Корпус типа 2140.20-1, масса не более 3,6 г.

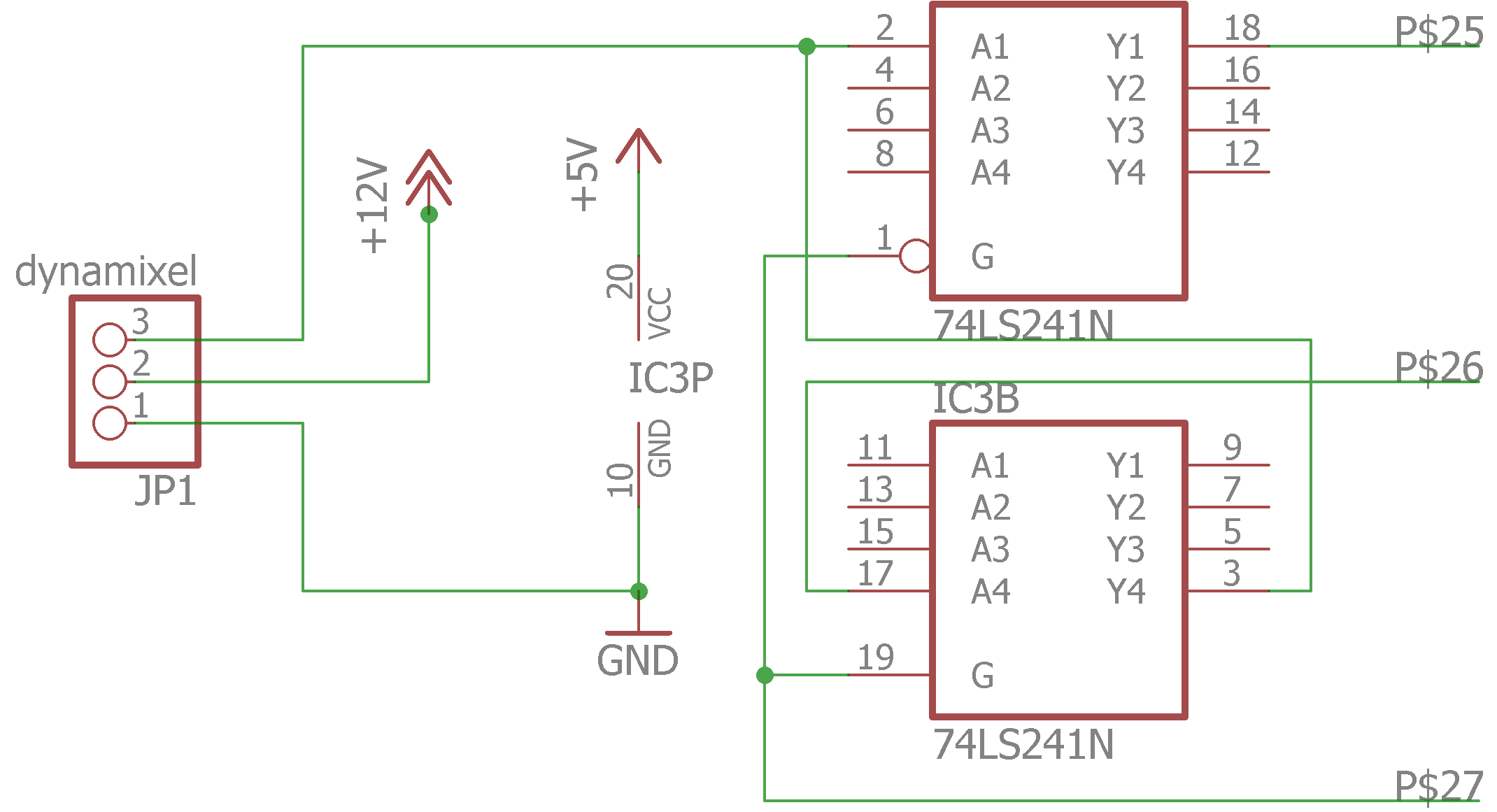


Рисунок 1. Схема подключения Микросхемы К555АП4.

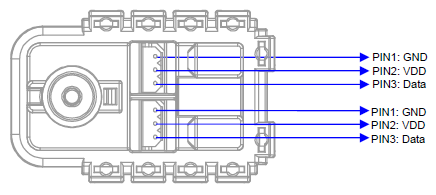


Рисунок 1. Схема распиновки сервомотора dynamixel AX-12a.

Таблица 1. Описание выводов платы световой индикации (фонаря).

|  |  |
| --- | --- |
| Обозначение | Наименование |
| VDD | положительный контакт питания 9-12 В |
| GND | отрицательный контакт питания |
| Data | цифровое управление, приём и передача данных |

* + 1. Датчик напряжения аккумулятора

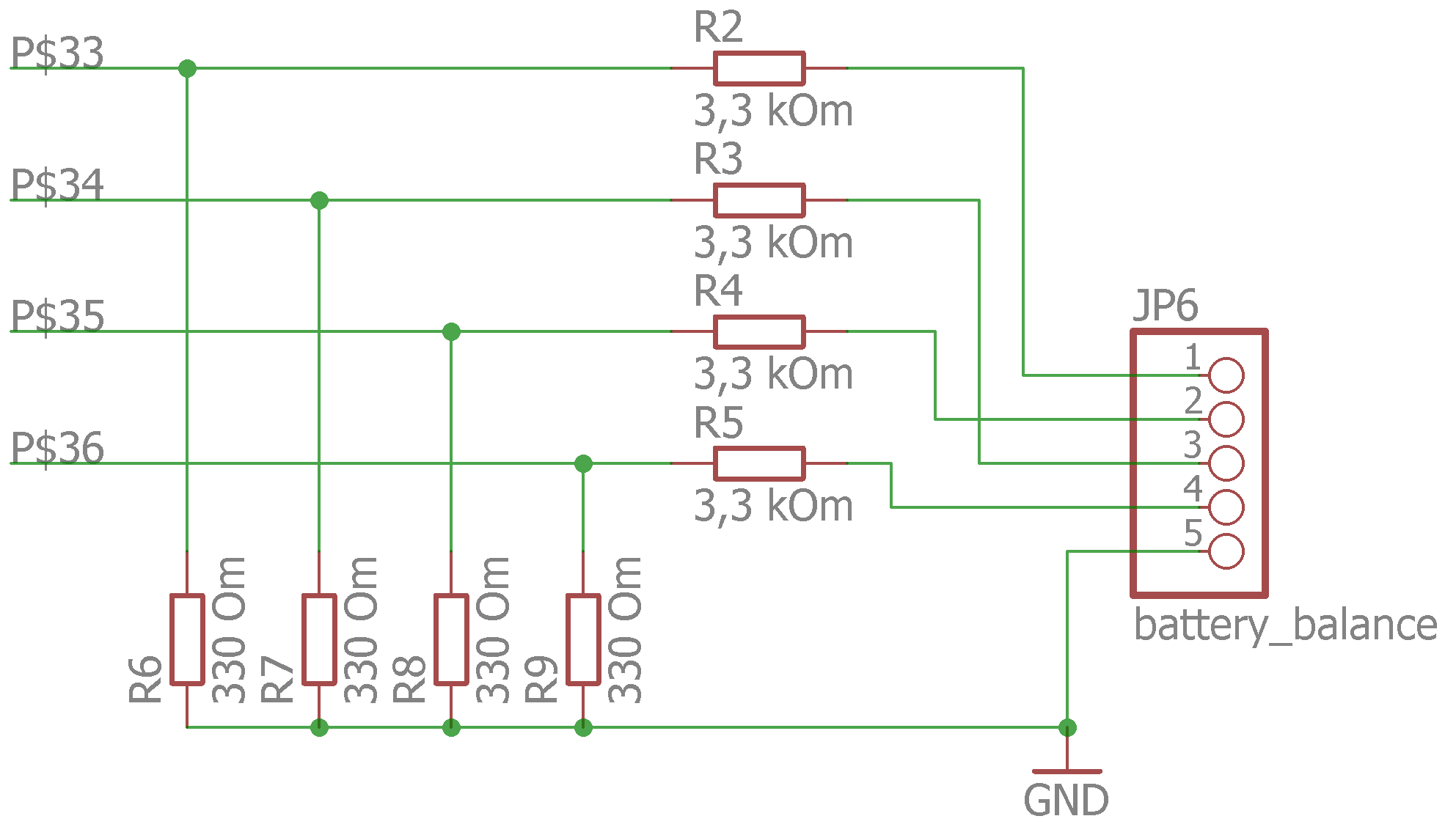


Рисунок 1. Схема датчика заряда аккумулятора к контроллеру.

* + 1. Датчик определения направления движения

Для корректного расчёта одометрии необходимо знать направление движения транспортного средства. При движении на выходах драйвера управления электродвигателями постоянного тока L298N, предназначенных для подключения моторов появляется положительное и отрицательное питание, которое меняется при изменении направления вращения колеса.

Для преобразования сигнала к дискретному с уровнем 5В с драйвера используется LM393P, Двойной маломощный компаратор напряжения.

Таблица 1. Технические характеристики LM393P.

|  |  |
| --- | --- |
| Наименование | Значение |
| Количество каналов | 2 |
| Напряжение питания | 4 – 30В |
| Время задержки | 300 нс |
| Ток потребления | 1 мА |
| Температурный диапазон | 0 – 70 °С |
| Тип корпуса | dip8 |
| Напряжение компенсации | 5 мВ |

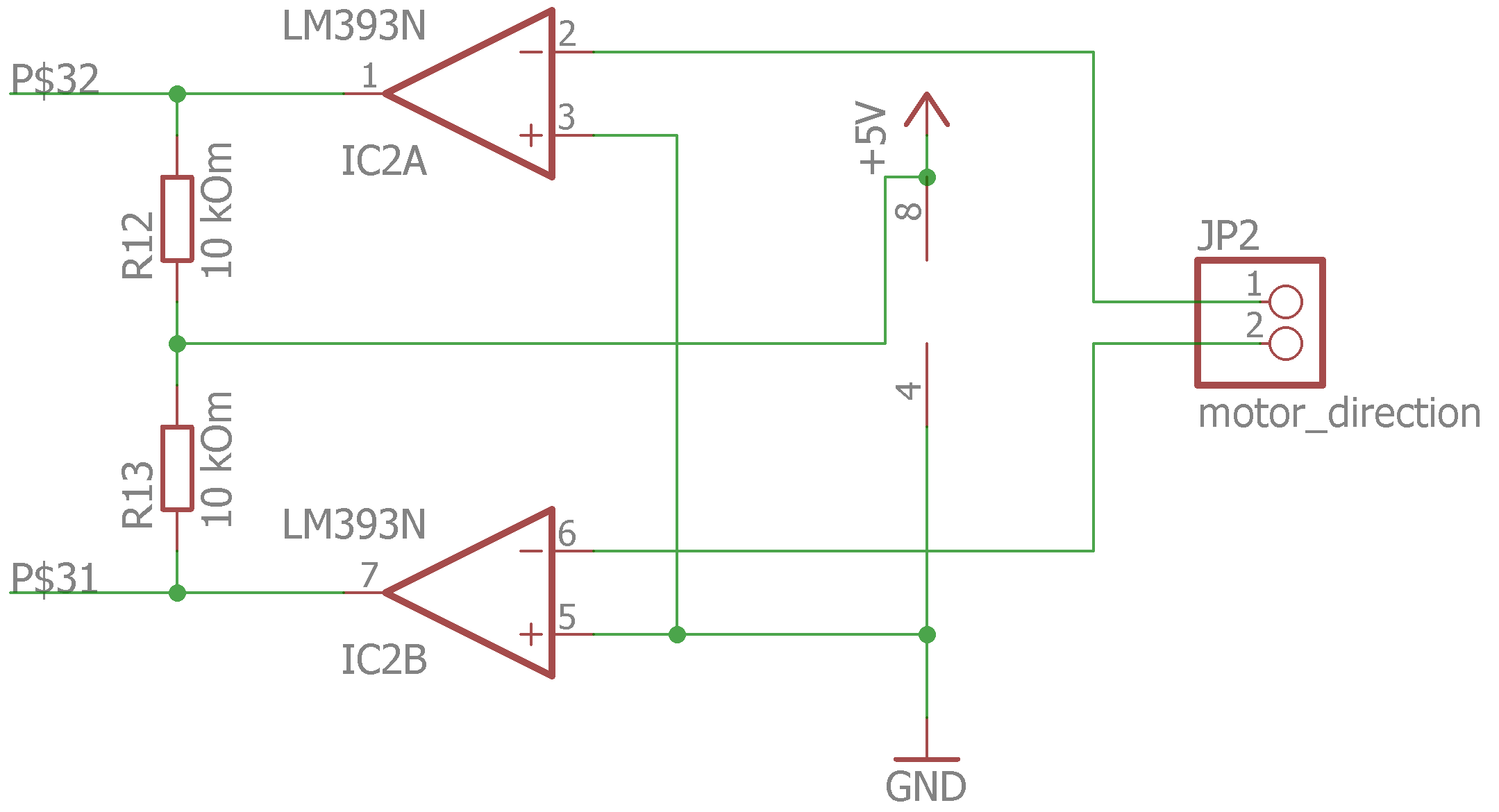


Рисунок 1. Схема подключения микросхемы LM393P к контроллеру

* + 1. Ультразвуковой дальномер

Для определения расстояния до препятствий относительно передней части автомобиля установлены два ультразвуковых дальномера HC-SR04.

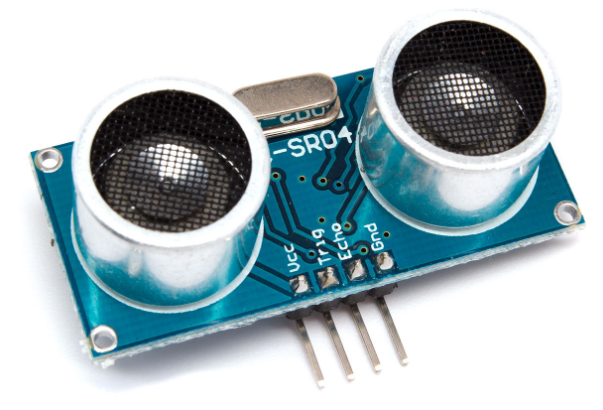
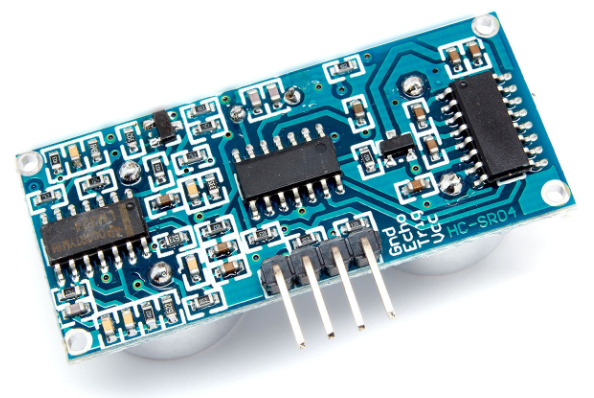
 

Рисунок 1. Ультразвуковой дальномер HC-SR04.

Таблица 1. Технические характеристики ультразвукового дальномера.

|  |  |
| --- | --- |
| Наименование | Значение |
| Напряжение питания | 5 В |
| Потребление в режиме тишины | 2 мА |
| Потребление при работе | 15 мА |
| Диапазон расстояний | 2 – 400 см |
| Эффективный угол наблюдения | 15° |
| Рабочий угол наблюдения | 30° |

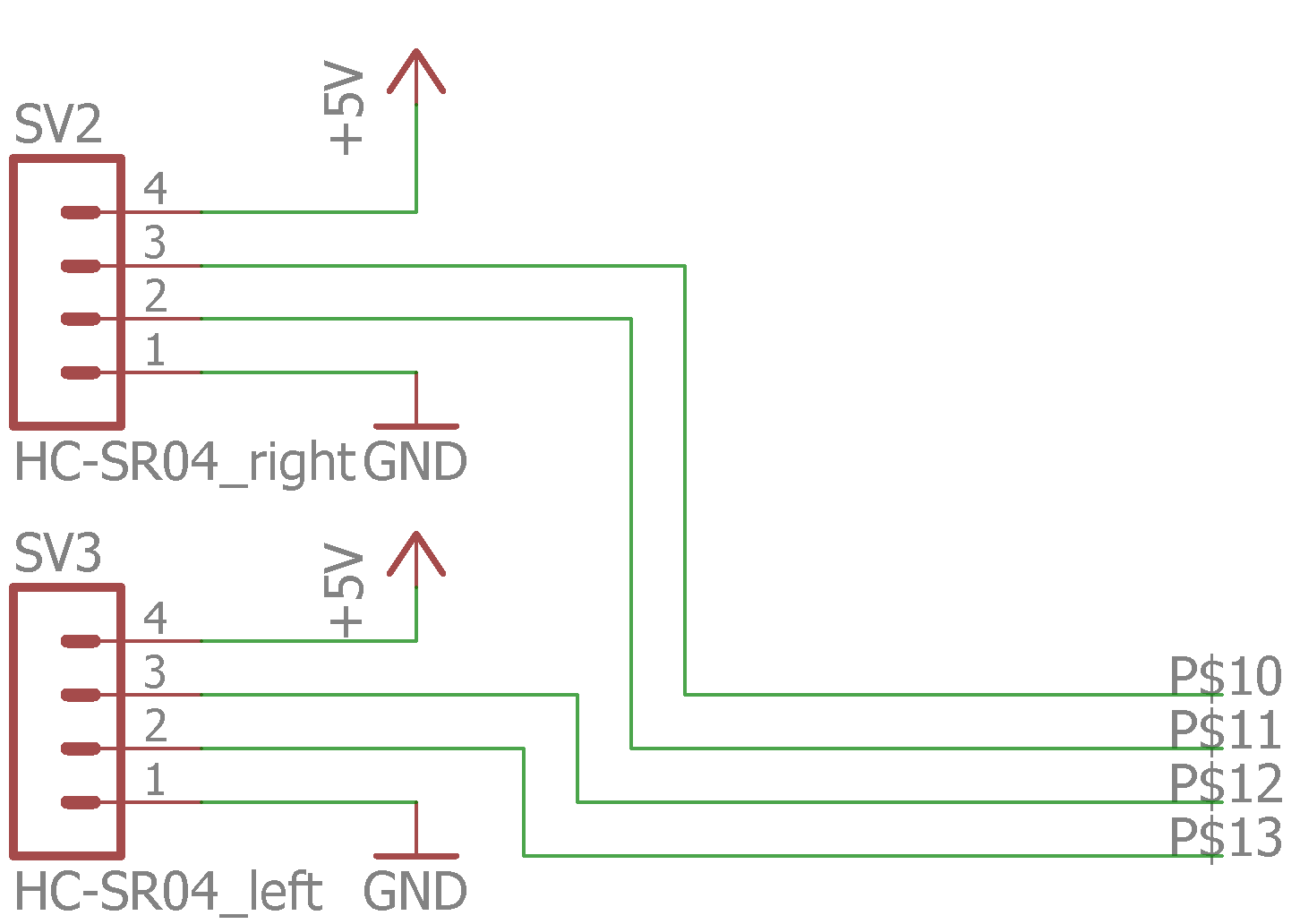


Рисунок 1. Электрическая схема подключения ультразвукового дальномера к контроллеру.

Таблица 1. Описание выводов ультразвукового дальномера.

|  |  |
| --- | --- |
| Обозначение | Наименование |
| Vcc | положительный контакт питания |
| Trig | цифровой вход. Для запуска измерения необходимо подать на этот вход логическую единицу на 10 мкс. Следующее измерение рекомендуется выполнять не ранее чем через 50 мс. |
| Echo | цифровой выход. После завершения измерения, на этот выход будет подана логическая единица на время, пропорциональное расстоянию до объекта |
| GND | отрицательный контакт питания |

* + 1. Световая индикация

Адресуемый светодиод – это RGB-светодиод, только с интегрированным контроллером WS2801 непосредственно на кристалле. Корпус светодиода выполнен в виде SMD компонента для поверхностного монтажа. Такой подход позволяет расположить светодиоды максимально близко друг другу, делая свечение более детализированным. Схема подключения адресных диодов приведена на рисунке ниже.



Рисунок 1. Адресный светодиод WS2818B.

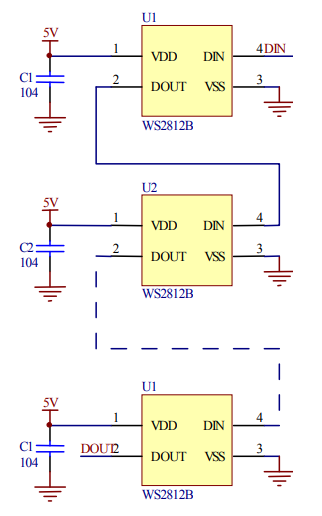


Рисунок 1. Схема подключения светодиодов WS2818B.

Таблица 1. Технические характеристики диодов WS2818B.

|  |  |
| --- | --- |
| Наименование | Значение |
| Питание светодиода | 3,5 ~ 5,3 В |
| Управление (вход) | -0,5 ~ +0,5 В |
| Температура эксплуатации | -25 ~ +80 ℃ |
| Температура хранения | -55 ~ +150 ℃ |

Фонарь представляет собой печатную плату с пятью светодиодами и необходимой для них обвязкой. На шасси располагается четыре фонаря, общее количество светодиодов: 20. Для экономии выводов с контроллера фонари подключаются последовательно. Адресация светодиодов начинается с нулевого элемента. Разработанная электрическая схема и печатная плата приведены на рисунках ниже. Печатная плата имеет крепежные отверстия для фиксации на бампере транспортного средства.

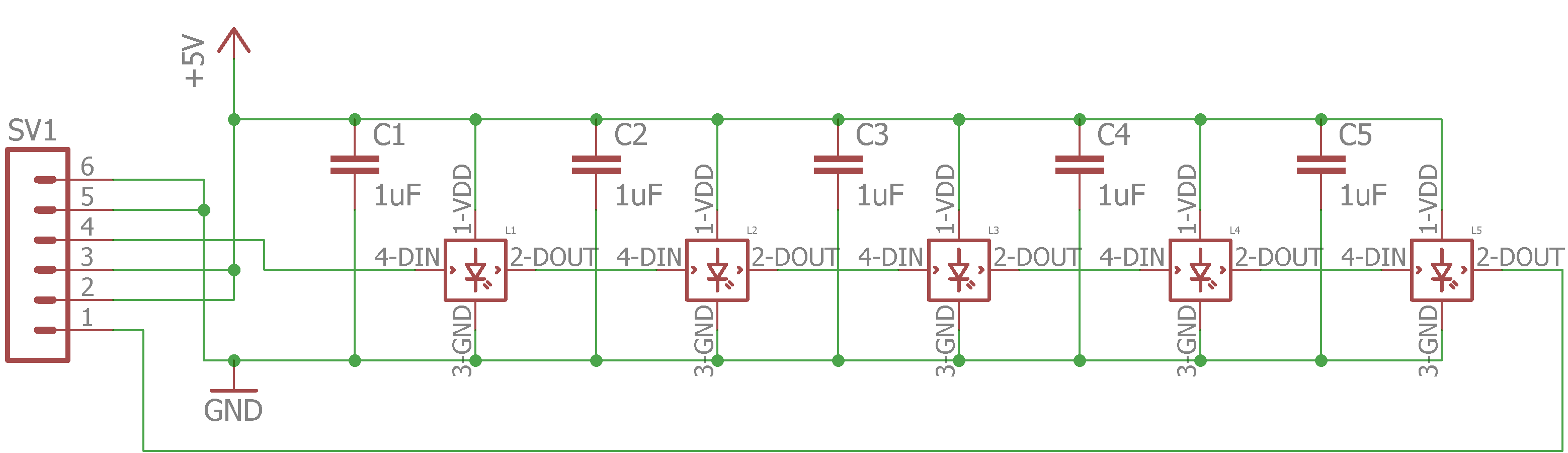


Рисунок 1. Электрическая схема световой индикации (фонарь).

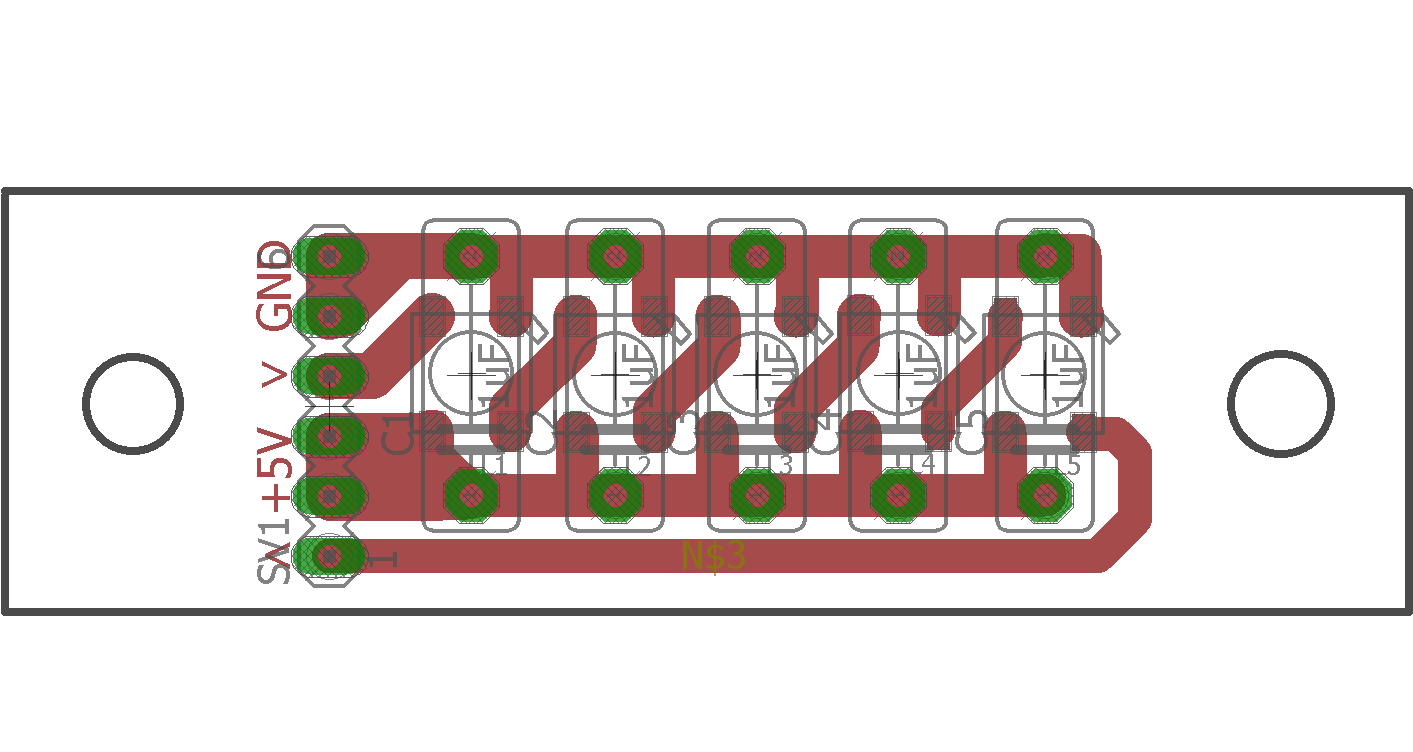


Рисунок 1. Плата световой индикации (фонарь).

Таблица 1. Описание выводов платы световой индикации (фонаря).

|  |  |
| --- | --- |
| Обозначение | Наименование |
| 5V | положитльный контакт питания 5В |
| GND | отрицательный контакт питания |
| > | вход управления |
| < | выход управления (для подключения следующего фонаря) |

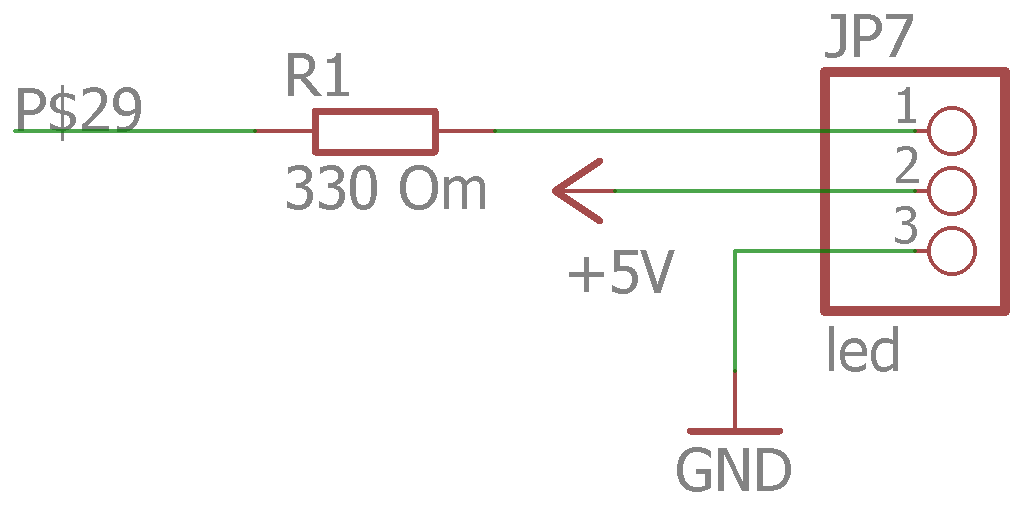


Рисунок 1. Электрическая схема подключения световой индикации к контроллеру.

3.3 Система управления высокого уровня

Общая информация

3.3.1 Микрокомпьютер

<http://wiki.t-firefly.com/en/Firefly-RK3399/>

<http://wiki.t-firefly.com/en/Firefly-RK3399/interface_definition.html#firefly-rk3399-interface-definition>

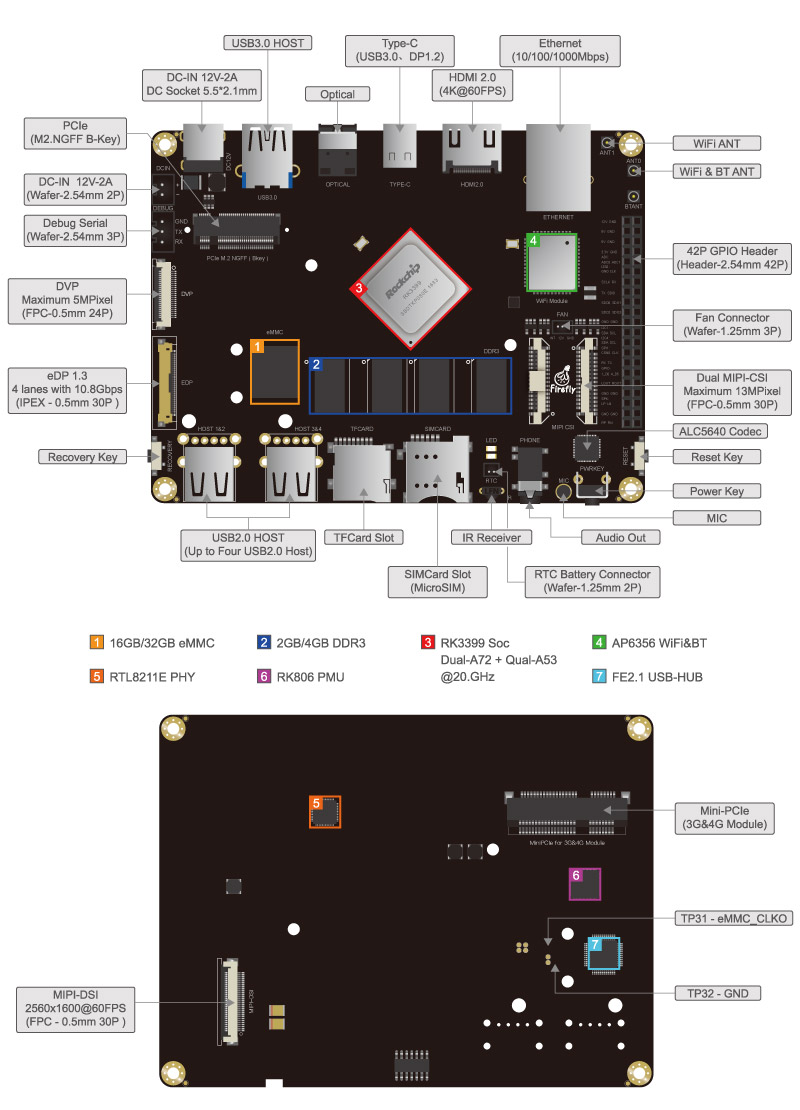


Рисунок 1. Описание компонентов микрокомпьютера.

3.3.2 Лазерный дальномер

Lidar LDS-01 – это лазерный 2D-сканер, способный измерять 360 градусов, собирая вокруг робота набор данных для использования для SLAM (одновременная локализация и картирование). Он поддерживает интерфейс USB и прост в установке на ПК. Принцип действия не имеет серьезных отличий от принципа работы радара: направленный луч источника излучения отражается от целей (непрозрачных тел), возвращается к источнику и улавливается высокочувствительным светочувствительным полупроводниковым прибором.



Рисунок 1. Лазерный дальномер LDS-01.

Таблица 1. Характеристики лазерного дальномера:

|  |  |
| --- | --- |
| Наименование | Характеристика |
| Рабочее напряжение | 5V DC ±5% |
| Источник | Полупроводниковый лазерный диод (= 785 нм) |
| Класс безопасности | IEC60825-1 Class 1 |
| Потребление тока | 400mA или меньше (Пиковый ток 1A) |
| Интерфейсы | 3.3V USART (230,400 bps) 42 байта на 6 градусов, опция Full Duplex |
| Сопротивление окружающему освещению | 10,000 люкс или меньше |
| Частота | 1.8kHz |
| Габаритные размеры | 69.5(В) X 95.5(Д) X 39.5(Ш)мм |
| Масса | Меньше 125г. |

Таблица 1. Характеристики измерений лазерного дальномера:

|  |  |
| --- | --- |
| Наименование | Характеристика |
| Диапазон | 120мм ~ 3,500мм |
| Точность (120-499)мм | ± 15мм |
| Точность (500-3500)мм | ± 5% |
| Прецизионность измерения (120-499)мм | ± 10мм |
| Прецизионность измерения (500-3500)мм | ± 3,5% |
| Скорость сканирования | 300 ± 10 об / мин |
| Угловой диапазон | 360 ° |
| Угловое разрешение | 1 ° |

Для подключения используется интерфейс USB2LDS, это плата, которая позволяет легко использовать лазерный дальномер с микрокомпьютером или ПК. Интерфейс USB2LDS подключается к микрокомпьютеру или ПК с помощью кабеля micro-USB и имеет порты для LDS и его двигателя.

Для реализации возможностей лазерного дальномера используется драйвер для ROS: hls\_lfcd\_lds\_driver.

3.3.3 Камеры

Инфа от Немировского

Глава 4. Программная реализация

4.1 Система управления низкого уровня (драйвер шасси)

4.1.1 Контроллер управления

Система управления низкого уровня реализована на контроллере STM32F103C8T6 (Blue Pill).

Для использования контроллера его необходимо прошить, используя USB-UART преобразователь. Для этого нужно переключить преобразователь в режим 3,3V и подключить выводы согласно таблице ниже.

Таблица 1. Подключение контактов USB-UART преобразователя к плате Blue Pill.

|  |  |
| --- | --- |
| USB-UART преобразователь | Blue Pill |
| Gnd | Gnd |
| Vcc | 3.3V |
| RX | PA9 |
| TX | PA10 |

Для программирования и загрузке скетча используя Arduino IDE необходимо добавить поддержку SMT32F103 (File / Preferences / Additional Boards Manager URL):

http://dan.drown.org/stm32duino/package\_STM32duino\_index.json

В Boards Manager (Tools / Board / Boards Manager) установить "SMT32F1xx/GD32F1xx boards by smt32duino" и активировать загрузку прошивки через Serial (Tools / Upload method / Serial). На плате нужно переключить перемычку BOOT0 в High, а BOOT1 в Low. После этого нужно произвести сброс платы и можно заливать скетчи. Для нормальной работы прошивки BOOT0 и BOOT1 должны быть в состоянии Low.

Затем в консоли требуется выполнить следующие команды:

$ wget -P /tmp https://github.com/rogerclarkmelbourne/STM32duino-bootloader/raw/master/binaries/generic\_boot20\_pc13.bin

$ cd ~/.arduino15/packages/stm32duino/tools/stm32tools/2018.12.3/linux /stm32flash

$ ./stm32flash -w /tmp/generic\_boot20\_pc13.bin -v -g 0x0 /dev/ttyACM0

После выполнения команд необходимо отключить USB-UART преобразователь, выставить перемычки BOOT0 и BOOT1 в Low и подключить плату к компьютеру через USB. Затем в Arduino IDE выбирается тип загрузки "STM32duino bootloader" (Tools / Upload method / STM32duino bootloader).

Последовательный порт платы опознается в Linux как /dev/ttyACM0 и его необходимо выбрать в Arduino IDE (Tools / Port).

Первую загрузку скетча нужно выполнять, включив режим infinite в загрузчике. Для этого подключается резистор номиналом 10k между 3,3v и PC14. После сброса светодиод на плате начинает постоянно мигать, что означает что загрузчик будет ждать начала загрузки без ограничения по времени. После первой загрузки этот режим можно выключить и дальше пользоваться платой как обычно.

Язык программирования Arduino основан на C/C++. Для использования ROS с платой Arduino используется библиотека — ros\_lib\_stm32.

ROS (Robot Operating System) предоставляет библиотеки и инструменты, помогающие разработчикам программного обеспечения создавать приложения для роботов. Он обеспечивает аппаратную абстракцию, драйверы устройств, библиотеки, визуализаторы, передачу сообщений, управление пакетами и многое другое. ROS лицензируется под открытым исходным кодом, лицензия BSD.

4.1.2 Управление электродвигателями

Узел подписан на топик cmd\_vel, сообщения имеют тип данных geometry\_msgs::Twist. В переменной cmd\_vel.linear.x содержится значение линейной скорости (м/с).

В функции linear2driverMotor(float linear\_speed) реализован ПИД-регулятор позволяющий поддерживать постоянную скорость используя среднее значение с оптических датчиков числа оборотов колес.

Исходный код позволяет изменять коэффициенты для ПИД-регулятора, значения содержатся в следующих константах: Kp, Ki, Kd.

Формулы ПИД-регулятора:

P = Kp \* e;

I = I\_prev + Ki \* e;

D = Kd \* (e - e\_prev);

motor\_value = round(P + I + D);

4.1.3 Рулевое управление

Для управления сервомотором используется библиотека DynamixelSerial3. В библиотеке реализован необходимый функционал для управления через третий последовательный порт с сервомотором.

В исходном коде реализована возможность корректировки идентификатора сервомотора (DYNAMIXEL\_ID), центрального положения колес (DYNAMIXEL\_CENTER), и максимального угла поворота колес (GRAD\_RUDDER).

В переменной cmd\_vel.angular.z содержится значение угла поворота колес в единицах угловой скорости (рад/с).

В функции angular2dynamixel(float angular) реализовано математическое преобразование угла поворота колес к значениям необходимым для управления сервомотором с программным ограничением максимального возможного угла поворота сервомотора относительно значений указанных в градусах (GRAD\_RUDDER).

Перевод единиц измерений (рад/с в градусы) производим по следующей формуле:

angular\_grad = angular \* 180.0 / M\_PI;

Расчёт максимального возможного значения поворота колес в одном направлении для управления сервомотором рассчитывается по следующей формуле:

angular\_value\_res = ((1024.0 / 300.0) \* (GRAD\_RUDDER));

Для расчета значения необходимого для управления сервомотором используется функция map и значения лимитов для сервомотора относительно центрального программируемого значения (DYNAMIXEL\_CENTER):

map(angular\_grad \* 100.0, -(GRAD\_RUDDER) \* 100.0, (GRAD\_RUDDER) \* 100.0, round(DYNAMIXEL\_CENTER - angular\_value\_res), round(DYNAMIXEL\_CENTER + angular\_value\_res));

4.1.4 Оптический датчик числа оборотов колеса и датчик направления движения

ляляляляляля

4.1.5 Датчик напряжения аккумулятора

Аккумулятор имеет балансировочный разъем, с помощью которого можно получать данные о напряжении на каждой секции аккумулятора. Информация о напряжении получается путем считывания значения с аналогового входа и пересчитывается с помощью экспериментально полученной функции. Ниже приведен график показывающий зависимость напряжения и значений аналового входа.

Рисунок 1. График соотношения напряжения и значений аналогового входа.

Коэффициент детерминации равный 0,9998, указывает высоком уровне достоверности аппроксимации.

Линия тренда имеет следующее уравнение:

V = 0.0088316807 \* analog + 0.031247058.

Значение напряжения 2, 3 и 4 секций аккумулятора получается за счет разности значений текущей секции с предыдущей. Общее значение напряжение аккумулятора рассчитывается путем сложения всех значений напряжения по секциям.

Информация о напряжении публикуется в топик battery, типом сообщений является sensor\_msgs::BatteryState.

4.1.6 Ультразвуковой дальномер

Данные о расстоянии до объектов получаемые от ультразвуковых датчиков, подвергаются фильтрации и публикуются в топики range/front/left\_snr и range/front/right\_snr, типом данных сообщений является sensor\_msgs::Range.

Фильтрация сигналов с ультразвуковых датчиков имеет важное значение, так как сигналы с датчиков имеют большую зашумленность, чтобы использовать их непосредственно. Чтобы исправить это используется фильтрация Калмана для оценки расстояния до объектов. Фильтр Калмана имеет еще одну особенность которая позволяет прогнозировать будущее значение.

Использовать ультразвуковые дальномеры позволяет библиотека Ultrasonic, а для фильтрации данных получаемых от дальномеров используется библиотека SimpleKalmanFilter.

4.1.7 Световая индикация

4.2 Система управления высокого уровня

В основном ROS используется в связке с микроконтроллерами, которые обладают большими вычислительными возможностями и собственной операционной системой.

Система управления высокого уровня реализована с помощью ROS.

4.2.1 Узел hls\_lfcd\_lds\_driver

Использовать лазерный дальномер позволяет пакет ROS для LDS (HLS-LFCD2). LDS (Laser Distance Sensor) – это датчик, отправляющий данные на хост для одновременной локализации и картирования (SLAM). Одновременно данные об обнаружении препятствий также могут быть отправлены на хост. Компания HLDS (Hitachi-LG Data Storage) разрабатывает технологию для датчика подвижной платформы.

Информация с лазерного дальномера публикуется в топик lidar\_scan, типом данных сообщений является sensor\_msgs::LaserScan, представляющий собой множество точек в пространстве, для каждой точки вычислен угол этой точки относительно начала отсчета.

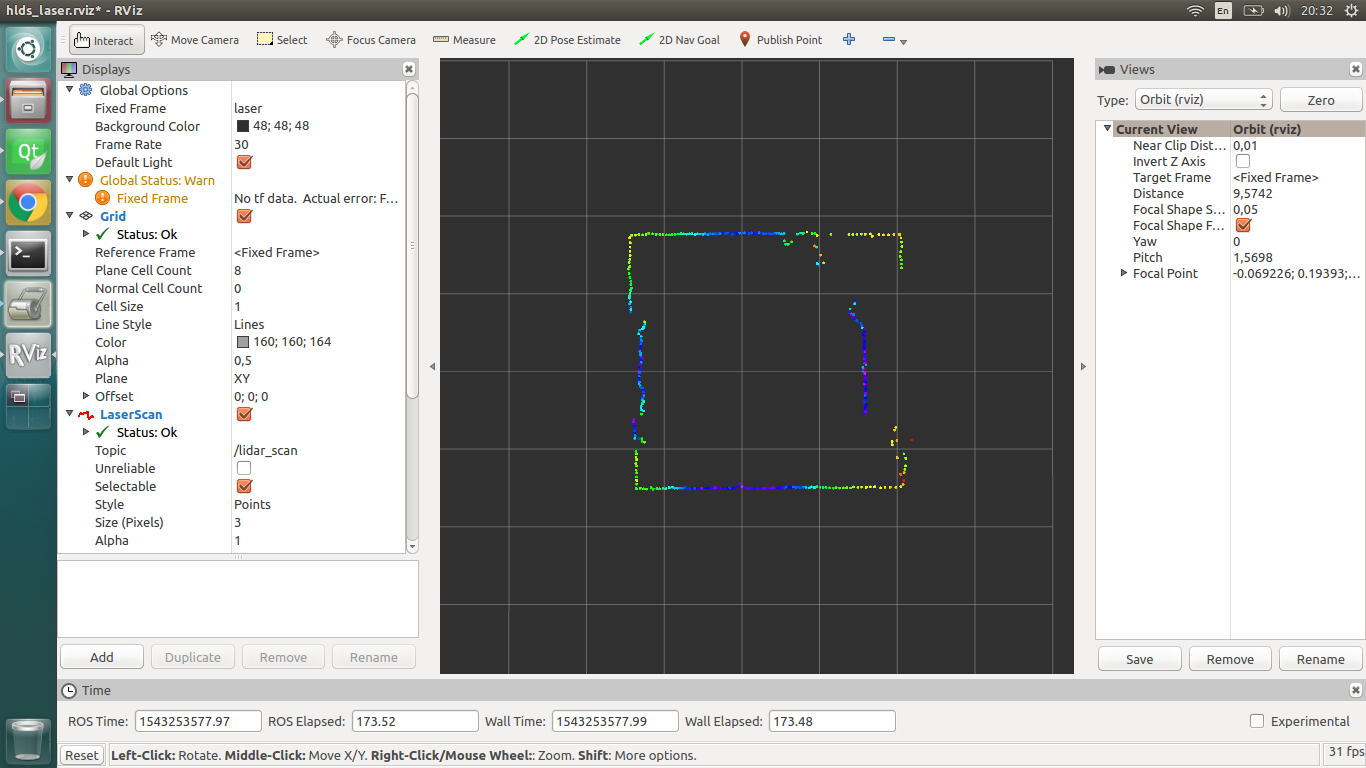


Рисунок 1. Визуальное представление данных получаемых с лазерного дальномера.

На рисунке изображено визуальное представление некоторого помещения с препятствиями. По цветовой температуре визуально видно на сколько далеко находится каждая точка относительно лазерного дальномера.

Установка:

$ sudo apt-get install ros-kinetic-hls-lfcd-lds-driver.

Предоставление прав на порт:

$ sudo chmod a+rw /dev/ttyUSB0.

Запуск узла:

$ roslaunch hls\_lfcd\_lds\_driver hlds\_laser\_segment.launch.

Загрузка драйвера и исходного кода:

$ git clone https://github.com/ROBOTIS-GIT/hls\_lfcd\_lds\_driver.git.

Launch файл включает в себя следующий код:

<?xml version="1.0"?>

<launch>

<node pkg="hls\_lfcd\_lds\_driver" type="hlds\_laser\_segment\_publisher" name="hlds\_laser\_segment\_publisher" output="screen">

<param name="port" value="/dev/ttyUSB0"/>

<param name="frame\_id" value="laser"/>

</node>

</launch>

4.2.2 Узел odometry

Одометрия — использование данных о движении приводов, для оценки перемещения транспортного средства.

Узел имеет подписку на топик joint\_states, сообщения имеют тип данных sensor\_msgs::JointState. В сообщениях имеется информация о текущей скорости вращения и количестве оборотов задних колес, а также угол поворота передних колес.

После математического пересчета информация о перемещении транспортного средства публикуется в топик odom, типом данных сообщений является nav\_msgs::Odometry.

Информация в топиках обновляется с частотой 50 герц.

Запуск узла:

$ roslaunch odometry odometry.launch.

Launch файл включает в себя следующий код:

<?xml version="1.0"?>

<launch>

<node pkg="odometry" type="odometry\_node" name="odometry\_node" respawn="true" output="screen">

</node>

</launch>

Respawn / Required являются не обязательными аргументами, эти аргументы используются по отдельности и никогда не используются вместе. Если respawn = true, то этот конкретный узел будет перезапущен, если по какой-то причине он закрылся. Если required = true то произойдет наоборот, то есть, если этот конкретный узел выйдет из строя, он закроет все узлы, связанные с файлом запуска.

4.2.3 Узел hector\_slam

Метод одновременной локализации и построения карты (SLAM от англ. simultaneous localization and mapping) — метод, используемый в мобильных автономных средствах для построения карты в неизвестном пространстве или для обновления карты в заранее известном пространстве с одновременным контролем текущего местоположения и пройденного пути. Популярные методы приближенного решения данной задачи включают в себя фильтр частиц и расширенный фильтр Калмана.

Узел hector\_slam для работы использует данные 2D лазерного дальномера в формате sensor\_msgs::LaserScan (публикуемые в топик /lidar\_scan) и выполняет построение карты и локализацию на той же частоте, с которой выполняется сканирование лазерным дальномером.

На данный момент в ROS имеются несколько алгоритмов SLAM. К сожалению, большая их часть не поддерживается в последних версиях ROS.

gmapping. Одна из самых популярных реализаций. Данный алгоритм использует данные, поступающие с лазерного дальномера и с энкодеров. Однако существую способы заменить лазерный дальномер на другой датчик расстояния, например Kinect.

hector\_slam. Данная реализация также является одной из самых популярных, а её основной особенностью считается возможность не использовать данные, полученные с одометрии робота.

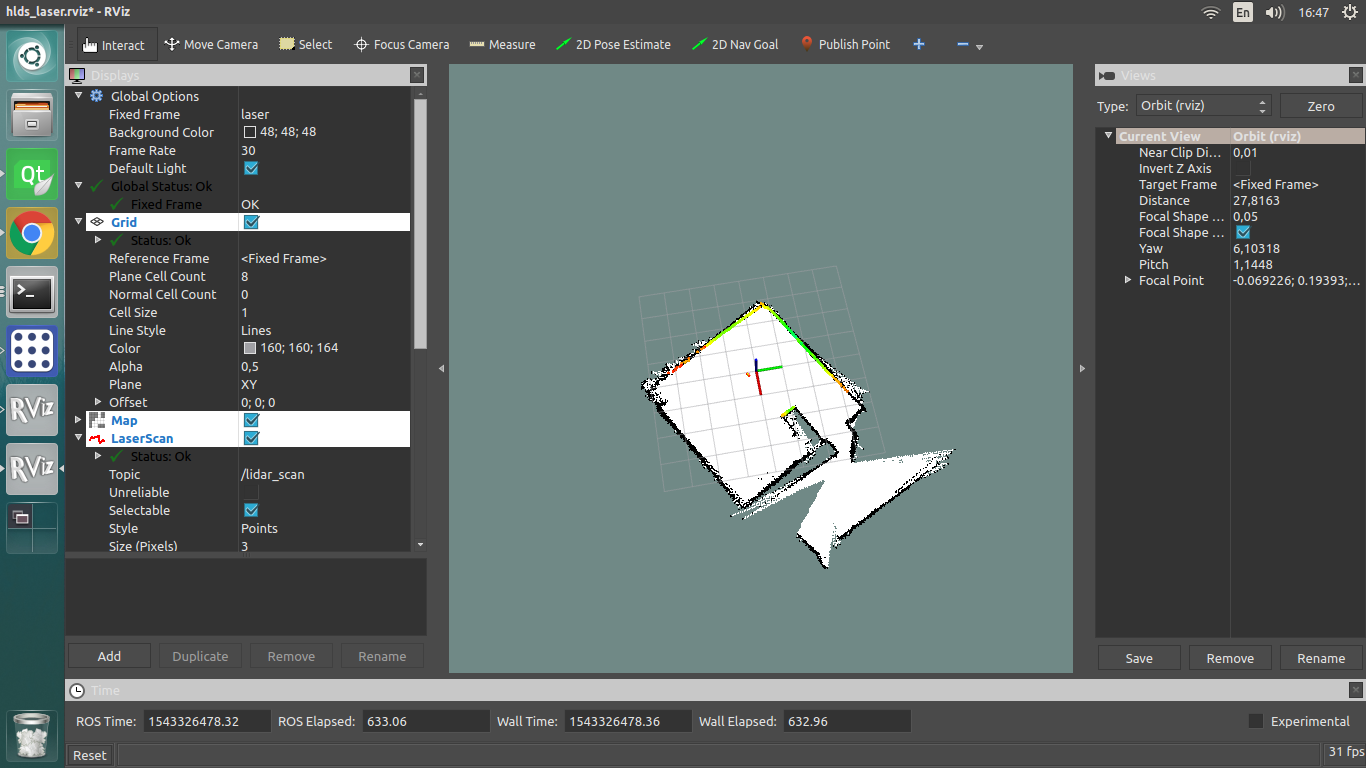


Рисунок 1. Карта, построенная при помощи SLAM.

Установка ROS пакет hector\_slam через apt-get:

$ sudo apt-get install ros-kinetic-hector-slam.

Загрузка драйвера и исходного кода:

$ git clone https://github.com/tu-darmstadt-ros-pkg/hector\_slam.