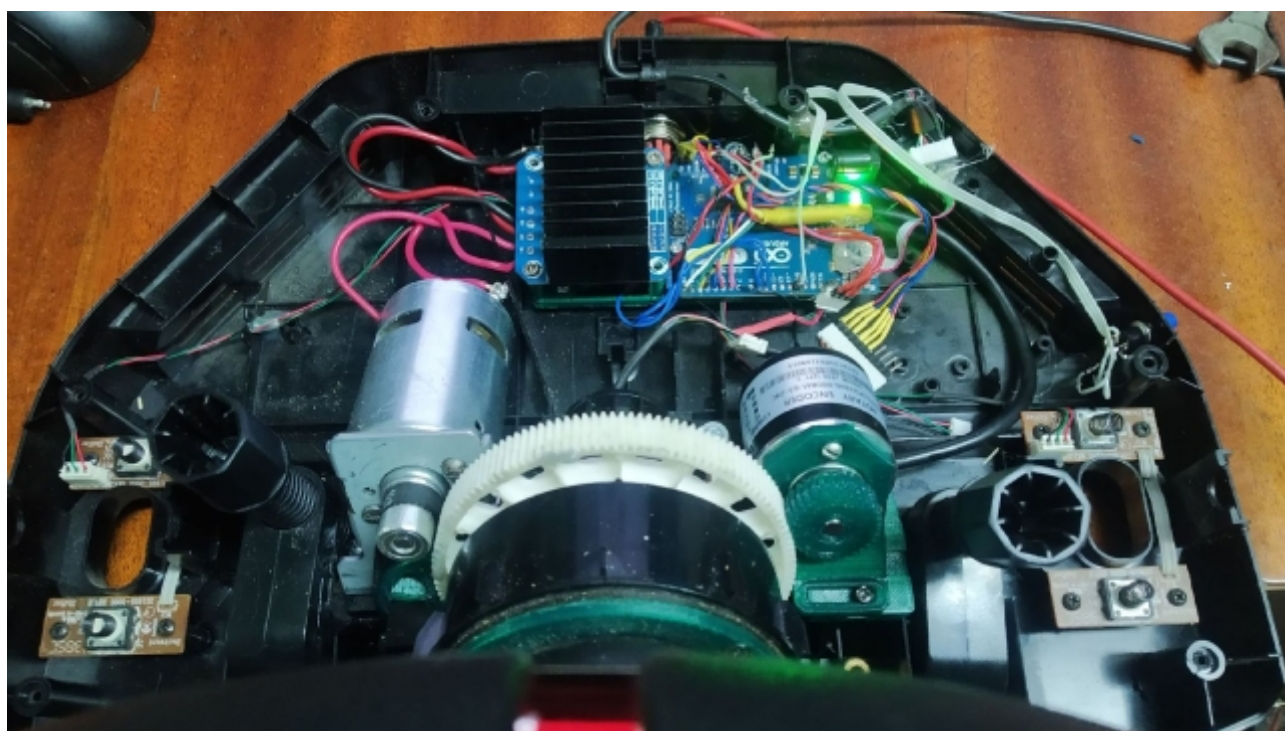


DIY G25

*ремонтируем,
модернизируем,
создаем
бюджетный руль
с FFB
своими руками*



JCD(JCDFCE) 2021г.

Оглавление

Вступление.....	3
Описание проекта.....	4
Электронные компоненты.....	5
Ардуино.....	5
Энкодер.....	5
Датчики педалей.....	5
Мотор.....	6
Драйвер.....	6
Блок питания.....	6
Структура архива:.....	7
Описание скетчей.....	7
Develops.....	7
Release.....	7
CalibrationG25.....	7
Установка Release.....	8
1 Подготовка.....	8
2 Инициализация EEPROM.....	8
3 Калибровка.....	8
Настройка оси руля.....	8
Калибровка педалей.....	8
4 Загрузка основного скетча.....	9
Легкий путь.....	9
Другой путь.....	9
Хаки и трюки.....	10
Инвертирование педали.....	10
Использование драйвера L298N.....	10
Схема руля МОМО с родным энкодером.....	10
Как вернуть порты ардуино?.....	10
Не заливается прошивка в ардуино через Xloader, что делать?.....	10
Как подключить USB-TTL конвертер?.....	10
Ссылки.....	11

Вступление.

Здравствуйте.

Этот проект - измененный скетч пользователя Sulako с форума schem.net на основе проекта MrShilov.

За этот титанический труд им огромная благодарность.

Оригинальный проект с которого все началось вы можете найти здесь

<https://cxem.net/mc/mc207.php>

К сожалению на просторах Интернет эти проекты малоизвестны, а существующие проекты с закрытым исходным кодом и платными прошивками/драйверами/функциями не способствуют развитию технической мысли.

Этот проект не идеален, многое можно переработать и улучшить, но с помощью этого проекта многие бюджетные рули можно улучшить или отремонтировать.

На данный момент магазины завалены рулями всех цен и категорий, но в связи с просадкой национальных валют, рули с активной обратной связью (ForceFeedback) и углом поворота 900 градусов стали недоступны многим из-за высокой цены, что делает интерес к DIY-рулям еще выше.

Описание проекта.

Плюсы проекта.

1. Открытость, в отличие от других проектов, этот проект имеет открытый код с MIT-лицензией, что позволяет его дорабатывать и изучать.
2. Доступная электроника, всё для проекта можно купить в интернет-магазинах или местных радиомагазинах, также частично можно использовать части (моторы, кнопки, энкодеры) ремонтируемых рулей.
3. Достаточное количество кнопок - 24 + 8(сверх дефолтных), три педали.
4. Использование драйвера от Logitech G25 делает совместимым руль с очень многими играми.
5. Отлично подходит для ремонта и модернизации многих бюджетных рулей
6. Работаящий FFB
7. Есть описание протокола FFB

Минусы проекта.

1. Некоторая сложность заливки скетча и прошивки.
2. Центровка кнопкой или установкой в центр до подключения руля.

Комментарий от Sulako к оригинальному скетчу.

Описание протокола FFB не во всем соответствует реальному поведению драйвера, и несколько неполное, так что пришлось кое-где от него отойти.

Не уверен, что результат правильный. К описанию эффектов есть вопросы, впрочем, реально в играх большая часть и не используется.

Не реализованы: эффект "Square Wave", управление светодиодами, watchdog, Raw Mode, расширенные команды, установка Dead Band, HID++ - либо оно ни к чему, либо описание поверхностное.

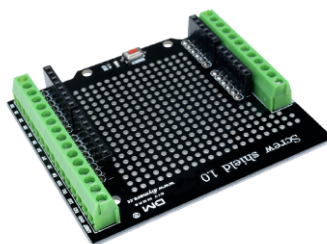
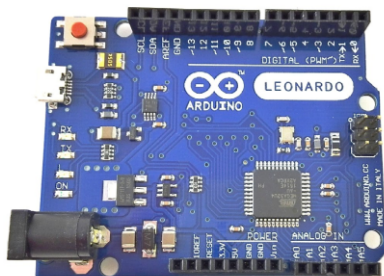
В качестве датчика оси руля используется оптический энкодер, датчики педалей - переменные резисторы или любые другие датчики с соответствующими цепями согласования и аналоговым выходом.

Электронные компоненты

Ардуино

Используется плата Ардуино Леонардо.

Рекомендуется использовать также шилд, чтобы облегчить подключения периферии.



Энкодер

Выбор огромен, рекомендую использовать стандартный энкодер руля, при его отсутствии лучше использовать промышленный энкодер.



При покупке обратите внимание на напряжение при котором он работает, нужен на 5Вольт, и количество тиков(импульсов) на оборот.

Если руль прямого привода (директ-драйв) и датчик на оси мотора, то следует брать энкодер с 1200 импульсами на оборот и выше (до 2400). Если руль с передачей (шестеренчатая, ременная) и энкодер стоит не на оси руля, то подбирайте энкодер таким образом чтобы общее количество импульсов было от 1200 до 2400 на 1 оборот

руля. Например, вал делает один оборот, а энкодер делает 4 оборота, то достаточно будет энкодера с 360-600 импульсами.

Не стремитесь выйти за 2400 тиков на один оборот руля, так как у Леонардо может не хватить скорости для подсчета импульсов на рывковых ускорениях.

Датчики педалей

Как обычно это переменные резисторы,

если улучшаете или ремонтируете руль, то они уже стоят в педалях. Обратите внимание, что нужны переменные резисторы с линейной характеристикой, у импортных в названии она обозначается буквой **В**.

По сопротивлению - лучше покупать переменные резисторы в диапазоне 10-20кОм.

Если у вас хорошие точные педали, есть смысл заменить резисторы на магниторезистивные промышленные датчики угла, типа такого

R3022-V1-CW360

Дорого, но зато надежно и долговечно.



Обратите внимание что им нужно питание 5В, и если используете в готовых педалях, то проверяйте полярность питания перед подключением.

Для серьезных педалей используются тензодатчики на 70-90кг, но не думаю что в этот проект не стоит с ними заморачиваться, это удел педалей к профессиональным базам.

Мотор

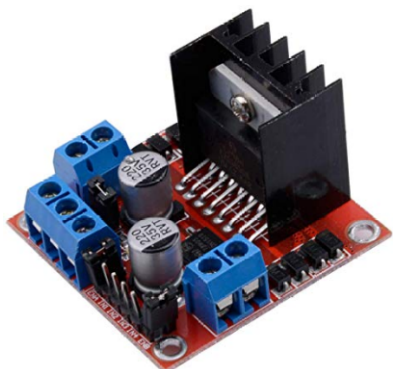
Мотор используется от самого руля или покупается 775-мотор на 3000-5000 оборотов.



Можно ли еще мощнее? Можно, но придется использовать более мощный драйвер, даже при установке 775 в MOMO Racing благодаря комбинированной передаче руль становится очень тяжелым при максимальной силе отдачи.

Драйвер

Если используете мотор из руля, то хватит драйвера типа L298N



Но для работы более крупного мотора лучше драйвер взять мощнее, например BTS7960



Обратите внимание, что управление драйверами разное, BTS7960 требует три линии управления, а L298N – две. При использовании двух линий для BTS7960 будет активирован режим торможения, что не позволит дрифтовать в гонках.

В оригинальном скетче Sulako использован L298N

В моем использован BTS7960, но вы можете использовать и L298N, подробнее об этом в разделе Хаки и трюки.

Блок питания

По опыту, если используете родной мотор руля, то и блок берете родной, если 775 мотор, то лучше использовать регулируемый блок с ограничением по току, на первое время напряжение 12В и ограничение по току в 2-3Ампера. Можно использовать DC/DC-модули с ограничением по току.

Структура архива:

- Original - оригинальный скетч Sulako
- Libraries - библиотеки для сборки скетча
- CalibrationG25 - скетч для предварительной настройки и калибровки перед прошивкой Release версии
- Develops- версия под энкодер с ручной установкой драйвера для разработки и отладки
- Release - релизная версия для окончательной загрузки
- Schematic - схемы устройств
- Bonus - дополнительные материалы, распиновки, документы.
- HEX – скомпилированные прошивки.

Описание скетчей.

Оригинальный скетч Sulako предназначен для резистора в оси руля, и его описание от автора вы найдете в папке проекта.

Develops

Особенности - энкодер, упрощенная калибровка оси руля, ручная калибровка педалей, калибровка осуществляется через Монитор порта, поддержка угла поворота в драйвере, ручная установка.

Для его установки воспользуйтесь инструкцией из оригинального проекта Sulako.

Или в два щелчка залейте скомпилированную прошивку Devel.hex из папки HEX с помощью программы Xloader, найдете ее в бонусе. С драйвером придется повозиться по инструкции Sulako.

Release

Собственно ради него все и затевалось.

Основное отличие от отладочной - безпроблемное определение и установка драйвера Logitech.

Имеет особенности - после прошивки этой версией - порты бутлоадера и Serial станут недоступны из среды ардуино.

Чтобы иметь возможность снова откалибровать педали в случае замены резисторов, в устройство добавлен софтовый COM-порт на выводах ISP-разъема к которому можно подключиться с помощью конвертера USB-TTL.

Так как это релизная версия, то убран вывод в порт многих данных отладки.

Также написан скетч для предварительной калибровки устройства до прошивки основным скетчем.

CalibrationG25

Скетч для калибровки осей руля и педалей до заливки скетча Release.

Он просто записывает в EEPROM данные калибровки.

Тоже самое можно сделать залив отладочную прошивку Devel.hex

Установка Release

1 Подготовка

Скетч не требует настройки, однако по умолчанию отключено ограничение поворота руля усилием двигателя, если оно требуется, то рекомендую сначала подобрать значение на Developers скетче, потом укажите значение и раскомментируйте строки в скетче Release и только потом его заливайте.

Прим. Если калибровка уже была выполнена на Developers скетче, можно сразу переходить к загрузке скетча Release.

Для начала загрузите в плату Leonardo скетч CalibrationG25.

Проверка железа

В Мониторе порта введите команду **info**

Будет выведена информация об оси руля и оси педалей.

Понажимайте на педали, покрутите рулем и понаблюдайте как меняются цифры.

При нажатии педали значение должно возрастать, при кручении руля влево от центра значения отрицательные, вправо положительные.

Если педаль инвертирована, то просто поменяйте провода на крайних выводах переменного резистора местами.

Если нет возможности инвертировать педаль, например при использовании магниторезистивных датчиков, то в разделе Хаки и трюки будет показано как инвертировать педаль в коде.

Остановите вывод информации в монитор порта снова понав **info**

2 Инициализация EEPROM

Запишем в память дефолтные значения командой **initMem**

Проверим командой **showCal**

Значения должны отличаться от -1.

3 Калибровка

Настройка оси руля.

Так как у многих возникало много вопросов, то я максимально упростил эту процедуру.

Поверните руль до упора против часовой стрелки (влево).

Введите команду **setTick**

На появившееся предложение, поверните руль по часовой стрелке (вправо) до упора и введите команду **end**

Калибровка педалей.

Все педали в исходном отжатом состоянии.

Вводим команды:

minA - запись состояния газа

minB - запись состояния тормоза

Если есть сцепление, то вводим **minC**

Нажимаем педаль газа до упора и вводим **maxA**
Нажимаем педаль тормоза до упора и вводим **maxB**
Если есть сцепление, то нажимаем педаль до упора и вводим **maxC**

Теперь проверяем калибровку командой **showCal**
Значения осей педалей газа, тормоза и сцепления(если есть педаль) должны отличаться от 0 и 1023
Значение Mode должно отличаться от 2400.

Если все нормально, то можно переходить к загрузке основного скетча Release

4 Загрузка основного скетча.

Я скомпилировал две прошивки, одна с ограничением угла поворотом мотора, вторая без этой функции. По-хорошему вы должны отладить эту функцию на develops -версии под свое железо.

Легкий путь.

Запускаете программу Xloader, указываете порт, прошивку и жмете прошить.
Если программа пишет failed, то перезапускаем ардуину быстрым двойным нажатием reset, быстро выбираем появившийся новый порт и ждем Upload.

Другой путь.

Через стандартную среду ардуино.
Для начала нам нужно в папке с Arduino в файле hardware/arduino/avr/boards.txt поменять значения следующих строк:

leonardo.build.vid=0x046D

leonardo.build.pid=0xC299

Необязательно, но тоже можно, чтобы руль назывался не "Arduino Leonardo":

leonardo.build.usb_product="G25 Racing Wheel USB"

Теперь нам нужно отключить COM-порты в файле hardware\avr\1.8.4\cores\arduino\USBdesc.h

для этого раскомментируйте строку `//#define CDC_DISABLED`

```
// Uncomment the following line or pass -DCDC_DISABLED to the compiler
// to disable CDC (serial console via USB).
// That's useful if you want to create a USB device (like a USB Boot Keyboard)
// that works even with problematic devices (like KVM switches).
// Keep in mind that with this change you'll have to use the Arduino's
// reset button to be able to flash it.
#define CDC_DISABLED
```

Все теперь можно запускать среду ардуино, выбрать плату Leonardo и прошивать плату.
После прошивки, установите драйвер соответствующий вашей ОС и вперед к новым горизонтам в играх.

Прим. Не забудьте после успешной прошивки вернуть файлы Arduino к стандартному виду.

Хаки и трюки.

Инвертирование педали

Пример для педали газа.

Находим на главной странице скетча строку

```
//Опрос осей
```

строка ниже

```
G25.axes[AXIS_ACC] → setValue(analogReadFast(PIN_ACC));
```

и есть то что нам нужно, меняем ее на

```
G25.axes[AXIS_ACC] → setValue(1023-analogReadFast(PIN_ACC));
```

Ниже идут две строки в которых читаются значения тормоза и сцепления, если надо инвертировать — изменяем строку аналогично.

Использование драйвера L298N

Подключение драйвера будет такое же как в оригинальном скетче Sulako, схема есть в оригинальном архиве.

Скопируйте файл motor.cpp из оригинального скетча Sulako в скетч Release, заменив существующий файл.

Схема руля МОМО с родным энкодером

Схема подключения родного энкодера в разделе Bonus, каталог МОМО

Также пример есть в папке Schematics

Как вернуть порты ардуино?

Вариантов несколько, использовать программатор или через Xloader залив Devel.hex

Не заливается прошивка в ардуино через Xloader, что делать?

Если при прошивке пишет failed, то то перезапускаем ардуину быстрым двойным нажатием reset, быстро выбираем появившийся новый порт и жмем Upload.

Как подключить USB-TTL конвертер?

Распиновка лежит в папке Schematics. Отключаем ардуино от USB. Соблюдаем полярность. Не забываем что соединение линий данных идет накрест RX-TX, TX-RX.

Ссылки

[Оригинальный проект](#)

[Группа по модификации руля MOMO в ВК](#)

[Ютуб-канал SeGord автора модов руля Logitech MOMO Racing](#)