

Разработка USB устройства ввода в Linux

Михаил Белкин

20 августа 2021 г.

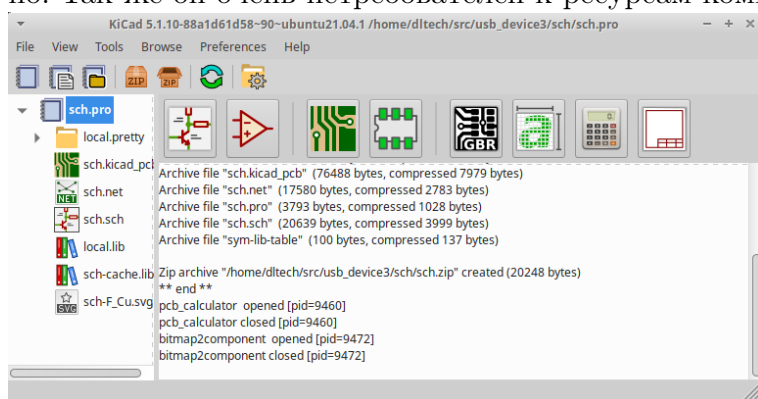
Содержание

1. Схемотехника.	2
1.1. Выбор редактора.	2
1.2. Установка редактора и библиотек.	2
1.3. Выбор элементной базы.	3
1.3.1. Микроконтроллер	3
1.3.2. Стабилизатор	3
1.3.3. Мелочь	4
1.4. Особенность схемотехники USB.	5
1.5. Схема устройства.	5
 2. Программное обеспечение.	 7
2.1. Тулчейн.	7
2.2. Компоновщик.	7
2.3. Что нового в CMSIS.	8
2.4. Структура проекта.	8
2.5. Опрос кнопок.	8
2.6. Периферия USB.	8
2.7. Дескрипторы USB.	8
2.8. Стандартный протокол USB.	8
2.9. USB HID	8
2.10. Отладка.	8

1. Схемотехника.

1.1. Выбор редактора.

Разработка схемы производится в KiCAD, это очень легковесный и компактный opensource редактор. Но при этом, несмотря на его внешнюю простоту, редактор как будто бы кричит нам "Я ничем не хуже чем этот ваш Altium и уж тем более Eagle". Поддерживается редактирование многослойных плат, так же используется профессиональный подход, при котором схема устройства и печатная плата редактируются отдельно. Так же он очень нетребователен к ресурсам компьютера.



К редактору имеется собственная библиотека компонентов, которая включает в себя компоненты всех популярных производителей. Но если вы принесли с китайского базара какую то экзотику, компонент придется разработать самостоятельно.

Вывод шаблона печатной платы возможен во всех удобных форматах, включая pdf. Так же и Gerber и векторный svg, последнее очень удобно для печати шаблона на принтере. Единственное неудобство это не возможность сразу выводить схему в растровом формате, приходится самостоятельно конвертировать из svg.

1.2. Установка редактора и библиотек.

Установка редактора в Ubuntu Linux производится очень просто, имеется отдельный ppa репозиторий с последней стабильной версией. В то время как наиболее полный набор библиотек компонентов можно скачать с гитхаба.

Набор команд для установки KiCAD:

```
sudo add-apt-repository —yes ppa:kicad/kicad-5.1-releases  
sudo apt update
```

```
sudo apt install —install—recommends kicad
```

Надеюсь устанавливать git вы умеете и про команду git clone вы тоже знаете. Вот ссылки на репозитории с библиотеками компонентов KiCAD:

<https://github.com/KiCad/kicad-library>

<https://github.com/KiCad/kicad-footprints>

<https://github.com/KiCad/kicad-symbols>

<https://github.com/KiCad/kicad-packages3D>

Установив редактор и добавив библиотеки вы сможете открыть проект со схемой устройства по адресу

1.3. Выбор элементной базы.

1.3.1. Микроконтроллер

Для наиболее аккуратной реализации нужен современный микроконтроллер (МК) с полноценным аппаратным USB. Совершенно понятно, что таким микроконтроллером окажется STM32F103C8T6. Мощное ядро ARM Cortex-M3 с их фирменным вложенным контроллером прерываний (NVIC) позволит с легкостью справиться с любой задачей. А с такой простой как USB геймпад уж тем более. На борту имеется 64 килобайта FLASH и 20 килобайт SRAM. И этого настолько много, что можно вовсе не думать об оптимизации. Теперь о стоимости, когда то я покупал такой за 60 рублей, сейчас цена приблизилась к 200, что по прежнему сравнимо по стоимости с остальными морально устаревшими микроконтроллерами. Так же в пользу данного микроконтроллера говорит наличие подробной документации. О том, почему именно F103, тут все просто, это самый дешевый МК с USB из тех что может предложить компания ST microelectronics.

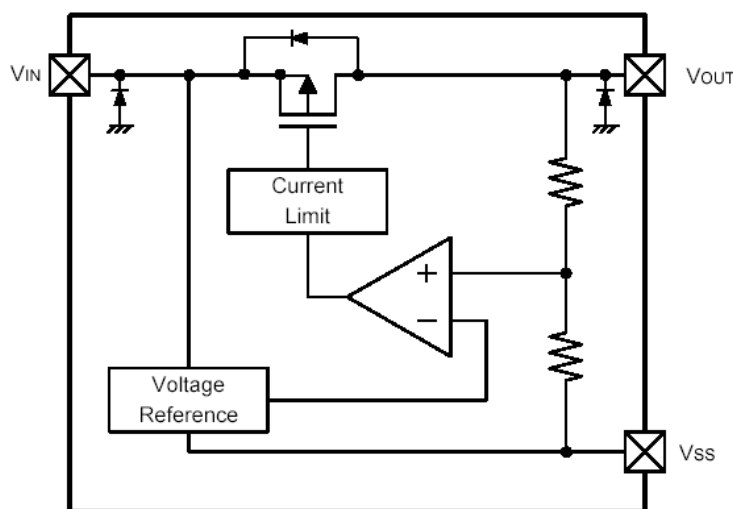
1.3.2. Стабилизатор

В шине USB, как известно, 5В, а номинальное напряжение питания МК 3.3В. Поэтому необходим понижающий стабилизатор напряжения. Я рассматривал три марки стабилизаторов. Они приведены в таблице ниже:

стаб	U_{inmax}	корпус	производитель	особенности
L78L33	30	SOT-89	ST microelectronics	
AMS1117-3.3	15	SOT-223	AMS semitech	термозащита
XC6206-33	7	SOT-23	TOREX	CMOS

И если первый давно знаком многим радиолюбителям. То последние два это стабилизаторы от китайских производителей, которые появились

недавно. В целом гораздо больше доверия к старому 78l, как минимум из за его большого входного напряжения. К тому же AMS1117 мне попадались нерабочими, и очень легко пробивались от скачков напряжения, не спасая нагрузку. Но хотелось бы компактней и подешевле, к тому же компьютер сам по себе стабильный источник питания. Поэтому я выбрал ХС6206. Довольно необычный новодел на полевых транзисторах, в то время как другие два на биполярных. Ниже приведена его структурная схема, на которой видны еще и защитные антистатические стабилитроны.



1.3.3. Мелочь

На форумах можно услышать совет не ставить кварц в целях экономии. Но в случае с асинхронной шиной кварц нужен для стабильной работы устройства. Микроконтроллер можно настроить на работу от самого распространенного кварца на 8МГц. Не удивительно, что на алиэкспресс сразу же нашелся не только планарный, но и очень компактный вариант. Размером как 1206 чип резистор.

Резисторы размером 0402 я успел заказать заранее. А вот шунтирующие конденсаторы пришлось выпаивать с донорских плат, потому что они не такие компактные, как хотелось бы (0805).

Расчетный ток потребления десятки миллиампер, потому для стабилизации питания хватит и чип керамики, благо такая есть даже на 10 мкФ. На всякий случай установлю токоограничивающий резистор по питанию. Основная его цель обезопасить компьютер от случайного короткого замыкания. Хотя в дорогих флешках на его месте можно встретить чип предохранитель. Продолжу экономить и на разъемах, попросту ограни-

чусь площадками под проводки.

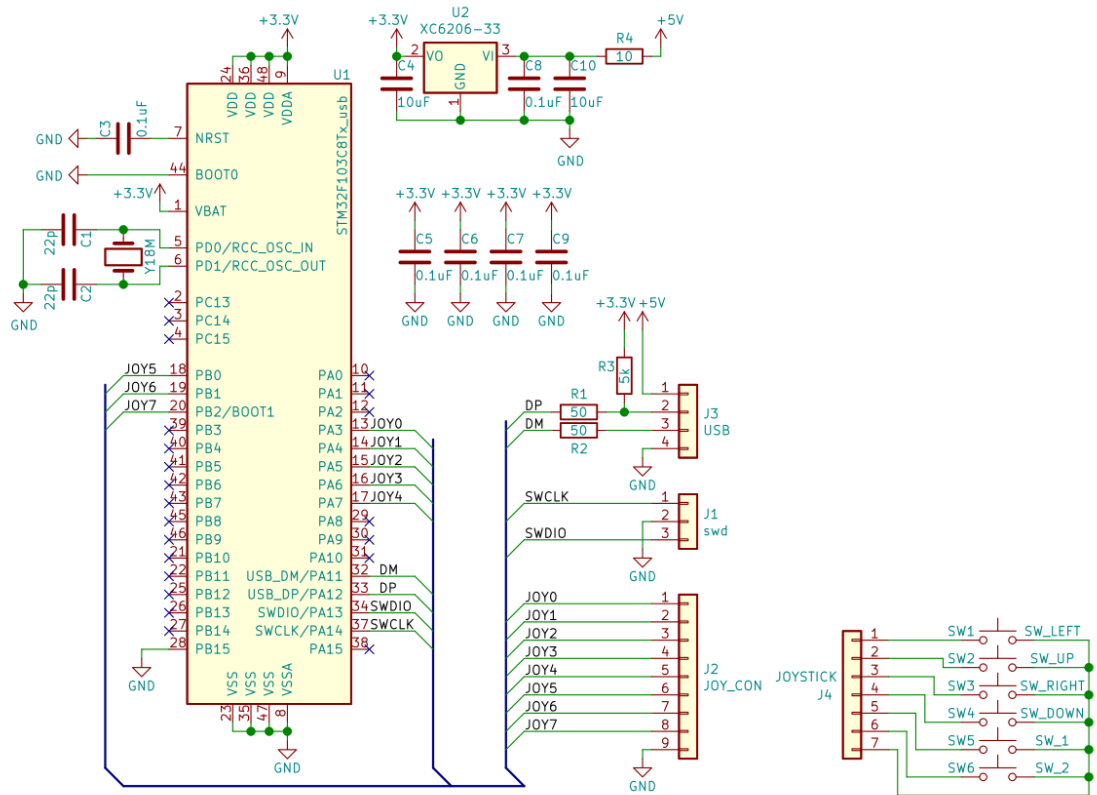
1.4. Особенность схемотехники USB.

На сайте можно скачать целый документ, посвященный вопросу распайки USB разъема. Основной вопрос заключается в возможности программного отключения устройства от ПК. Моё же устройство будет всегда включено, поэтому подтяжка линии DP к питанию будет постоянной, и осуществляться резистором, а не управляться транзистором и портом микроконтроллера. Также важно не забыть про защитные резисторы. Провод у меня используется готовый от клавиатуры, потому что разъем на плате не нужен.

1.5. Схема устройства.

А вот и схема целиком, как видите, все шины питания подключены и заземлены фильтрующими конденсаторами. Кварц с нагрузочными конденсаторами в наличии, так же подтянут к земле и порт сброса. Все как советует официальная документация. Кнопки джойстика, как видно, подключены к портам напрямую, т.к. внутри МК уже имеются резисторы подтяжки к 3.3В. Так же не обошлось и без так называемого "грязного хака", для упрощения трассировки платы один из портов ввода-вывода (28 вывод) использован как вывод земли. Но в этом нет ничего плохого, ведь порты после сброса находятся в состоянии с высоким входным сопротивлением.

Так же на схеме вы можете увидеть и стандартный разъем SWD для прошивки и отладки ПО. Ровно как и подключенный на землю вывод BOOT0 означает запуск прошивки из основной flash памяти.



2. Программное обеспечение.

Если по началу, когда эти чипы только появились. Выбора способов написания ПО для них было немного. Как правило это были коммерческие среды среды программирования под Windows, которые студенты и любители как правило скачивали в месте с так называемой таблеткой от жадности. Теперь же выбор бесплатных инструментов очень большой. Начиная с официальной среды от компании ST, STM32Cube. Которая может не только скачать сама все их библиотеки и примеры, но и имеет автоматический конфигуратор периферии (это такая штука, которая предлагает настраивать регистры тыкая мышкой). И заканчивая сторонними продуктами, таким как AC6 system workbench. Так же на гитхабе полно примеров сборки через Makefile. А также комьюнити создало полноценную библиотеку libopenm3 с кучей примеров.

Но исходя из того, что пример данного устройства довольно простой. Напишем весь код полностью самостоятельно.

2.1. Тулчейн.

Когда то просто скачивание и установка gcc была поводом для отдельной статьи. Я же отмечу только то, что gcc признает контроллеры на ядре Cortex M3. И в ubuntu устанавливается вот такой вот командой:

```
apt install gcc-arm-none-eabi libnewlib-arm-none-eabi gdb-multiarch  
apt install build-essential
```

А наши makefile и скрипты прошивки их будут их искать потом как arm-none-eabi-gcc, arm-none-eabi-objcopy и gdb-multiarch. В остальном даже флаги компиляции в случае с данными микроконтроллерами ничем особенно не отличаются.

2.2. Что нового в CMSIS.

2.3. Компоновщик.

Устройство довольно простое, почему бы по новой не написать целиком все. И так, не совсем целиком. Если игнорирование официальной библиотеки от производителя это уже немного глупо, то игнорировать библиотеки создателей процессора просто нельзя. И так, выбран процессор с ядром ARM Cortex M3, значит и будем опираться на файл CMSIS_5/blob/develop/Device/ARM/ARMCortexM3/Source/GCC/gcc_arm.ld.

- 2.4. Структура проекта.
- 2.5. Опрос кнопок.
- 2.6. Периферия USB.
- 2.7. Дескрипторы USB.
- 2.8. Стандартный протокол USB.
- 2.9. USB HID
- 2.10. Отладка.