Serial Interface

Me

September 12, 2016

Contents

1 Последовательный интерфейс

Мы будем использовать последовательный порт для вывода различной информации. Конечно выводить информацию можно и напрямую в видео память, т. е. упрощенно [1] на экран монитора. Но так как мы работаем с виртальной машиной QEMU и монитор соответсвенно тоже виртуальный, а значит пользоваться им не очень удобно. Было бы гораздо удобнее если бы вывод из виртальной машины попадал напрямую в терминал. QEMU позволяет перенправить информацию выведенную в последовательный порт в терминал, чем мы и будем пользоваться:

```
1 qemu-system-x86_64 -kernel kernel -serial stdio
```

Обратите внимание на опцию serila и ее аргумент stdio, эта опция и аргумент, собственно, и делают весь трюк.

1.1 Общая информация

Нас не интересует как физически передается информацию по последовательному интерфейсу, но логический уровень передачи мы все равно рассмотрим. Это поможет нам понять, по крайней мере частично, что и почему мы настраиваем.

Итак начнем с названия - последовательный. Последовательный значит, что биты сообщения передаются по одному каналу друг за другом. В противоположность параллельному интерфейсу, когда несколько бит передаются одновременно по нескольким параллельным каналам.

Кроме того биты группируются в фреймы. В дополнение к битам данных в фрейм могут быть добавлены стартовая и стоповая последовательности, которые позволят синхронизоваться приемнику и передатчику, а так же, например, бит проверки четности. Не удивительно, что приемник и передатчик должны использовать одинаковые настройки или осмысленной коммуникации не получится.

В нашем случае формат кадра определяют:

- количество бит данных в фрейме, возможные для нас опции это 5,6,7 или 8 бит данных; более или менее стандартный вариант 8 бит данных;

[1] Очень-очень-очень упрощенно.

- количество стоповых бит, возможные для нас варианты 1 или 2 (1,5); более или менее стандартная конфигурация это 1 стоповый бит;

- биты проверки четности, варианты включают проверку на четности, на нечетность или отсутствие каких-либо проверок, а так же кое-какие другие; более или менее стандратный вариант не использовать проверку четности вообще;

Кроме формата кадра, приемник и передатчик должны работать на одной и той же скорости. Скорость последовательного интерфейса обычно измеряют в бодах. Вообще бодами измерют символьную скорость, т. е. количество символов переданных в секунду. Один символ может нести больше одного бита информации, поэтому символьная скорость и обычная сокрость это не одно и то же, впрочем в нашем случае символом является как раз один бит, т. е. в нашем случае это одно и то же.

Скорость обычно задается коэффициентом деления. Т. е. контроллер последовательного связан с каким-то таймером, который с определнной частотой генерирует сигналы, на каждый такой сигнал приемник/передатчик принимает/передает один бит. Однако между таймером и контроллером последовательного интерфейса может находиться схема делитель частоты (prescaler). В зависимости от настроек схемы делителя, она может пропускать дальше только каждый n-ый сигнал от таймера, тем самым можно замедлить последовательный интерфейс.

Типичные используемые скорости: 9600 бот, 19200 бод, 38400 бод, 57600 бод и 115200 бод. Тут трудно выбрать какую-то более или менее типичную частоту.

1.2 Взаимодействие с последовательным интерфейсом

Контроллер последовательного интерфейса использует порты ввода/вывода, но их больше и их использование гораздо хитрее, чем, например, у контроллера прерываний intel 8259.

Наш котроллер последовательного порта использует порты начиная с 0x3f8, конкретные порты далее будут обозначаться в формате $+x$, что будет значить на самом деле порт $0x3f8 + x$.

1.3 Последовательный интерфейс и прерывания

Контроллер последовательного интерфейса IBM PC AT[1] может генерировать прерывания, а может и не генерировать в зависимости от настроек. Самый простой способ не использовать прерывания и опрашивать контроллер последовательного порта[2].

Чтобы отключить прерывания исползуйте порт +1. Разные биты в этом порте соответсвуют разным ситуациям, в которых контроллер последовательного порта может генерировать прерывания. Если мы хотим отключить прерывания нужно просто записать 0.

1.4 Задание скорости работы

Для задания скорости работы нужно задать коэффициент деления, он является двухбайтовым числом. Символьная скорость опредеяется коэффициентом деления по следующей простой формуле:

$$Speed = \frac{115200}{Div}$$

где

- *Speed* - символьная скорость;
- *Div* - коэффициент деления;

Для записи младшего байта коэффициента деления используйте порт +0, а для старшего байта используйте порт +1. И это не ошибка. Порт +1 используется и для задания режима генерации прерываний и для задания одного из байт коэффициента деления. Как же так? Отличить между этими двумя вариантами использования позволяет один из бит порта +3. Т. е. перед тем как задавать коэффициент деления вам сначала нужно записать правильное значение в порт +3, а перед тем как выключать прерывания нужно опять же записать в порт +3 значние, но уже другое. Подробности будут дальше, но если коротко, то если бит 7 (считая с 0) был установлен в значении записанном в порт +3, то при обращении к портам +0 и +0 мы задаем коэффициент деления, в противном случае нет.

Другими словами, перед заданием коэффициента деления запишите в +3 байт с установленным 7 битом, а затем запишите два байт коэффциента деления в +0 и +1 порты.

[1]Именно его мы и будем использовать.

[2]Это называется polling.

1.5 Задание формата кадра

Формат кадра задается опять же записью в порт +3. Мы уже знаем, за что отвечает 7-ой бит, хотя он и не имеет отношения к формату кадра. Теперь за что отвечают оставшиеся биты.

Младшие два бита задают количество бит данных в кадре. Возможных вариантов не много:

data bits	bit 1	bit 0
5	0	0
6	0	1
7	1	0
8	1	1

Следующий бит отвечает за количество стоповых бит, если 2-ой бит установлен, то используются 2(1,5) стоповых бита, в противном случае 1.

Биты с 3-его до 5-ого отвечают за проверку четности. Если 3-ий бит сброшен, то проверка четности не выполняется, если установлен, то режим проверки четности определяется двумя другими битами. Впрочем, учитывая, что проверка четности не особо нам нужна, то детали остаются на самостоятельно изучение по желанию.

Обратите внимание, что бит доступа к коэффициенту деления тоже доступен через порт +3, что значит, что если вы записываете в +3 значение с одним установленным 7-мым битом вы одновременно сбрасываете все остальное в 0, и наоборот, если вы задаете только формат кадра, то вы сбрасываете бит доступа к коэффициенту деления в 0.

1.6 Вывод информации в последовательный интерфейс

Наконец, осталось понять как заставить последовательный интерфейс передавать то, что мы хотим[1].

Для того, чтобы *запустить* передачу одного байта необходимо записать этот байт в порт +0. Обратите внимание, что если до этого вы записали в +3 значение с установленным битом 7, то запись в +0 изменяет коэффициент деления, а не запускает передачу данных, т. е. в последней записи в +3 7-ой бит должен быть сброшен, а одновременно с этим должен быть установлен формат кадра.

Но стоит обратить внимание на еще один важный момент. Уже было упомянуто, что последовательный интерфейс передает/принимает

[1]Получение информации нас не особо интересует - мы говорим, но не слушаем.

данные с некоторой фиксированной скоростью. Что произойдет, если мы будем пробовать передавать данные быстрее, чем последовательный интерфейс успевает их отсылать? Конечно, ничего хорошего. Поэтому нам нужно уметь определять, безопасно ли отслыать следующий байт данных. При использовании прерываний, это не стало бы проблемой, т. е. когда контроллер готов передавать следующую порцию данных будет сгенерированно прерывание, но без использования прерываний требуется делать как-то подругому.

Определить можно ли передавать следующий байт можно прочитав порт +5, а в частности нас интересует бит 5 (считая с 0). Если этот бит сброшен, то передавать еще нельзя и нужно подождать. Т. е. перед передачей очередного байта вы должны в цикле проверять, не установлен сброшен ли бит 5 или нет.

1.7 Вывод

После прочтения этого материала вы должны уметь настроить контроллер последовательного интерфейса и уметь передавать по нему информацию в polling режиме, т. е. без использования прерываний.