

Самый эффективный способ подключения инкрементального энкодера к МК

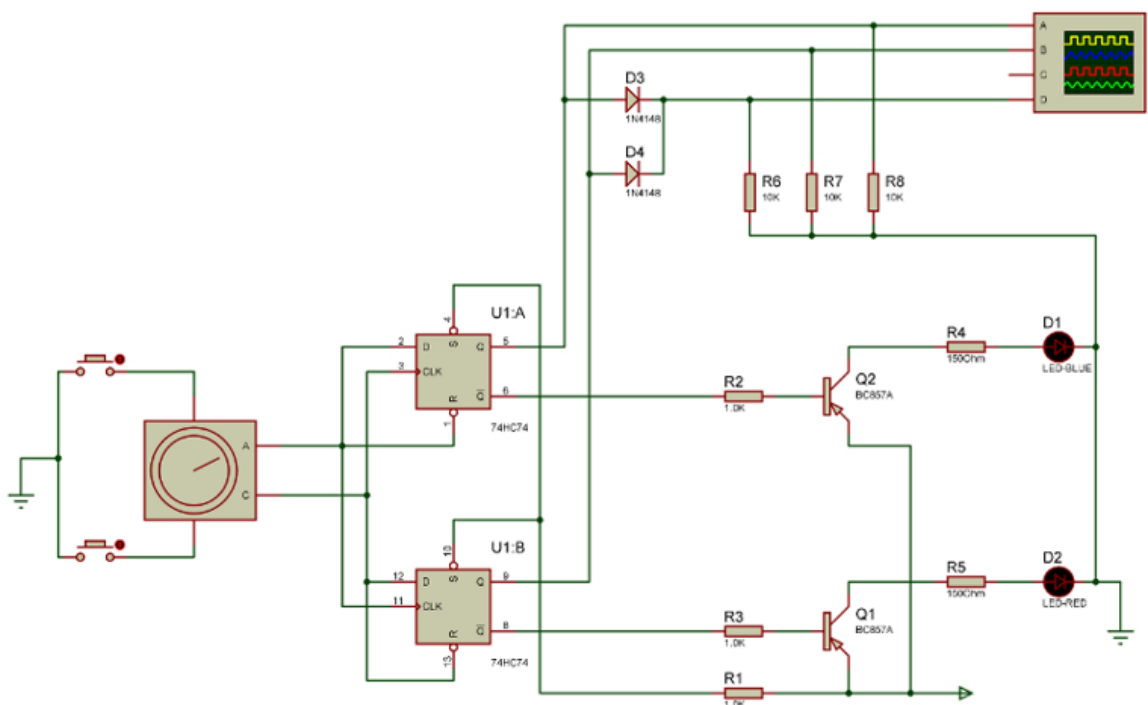
Плюсы :

- 1) Не требуется программной обработки дребезга контактов
- 2) МК обрабатывает энкодер только по прерыванию, а значит в главном цикле не требуется его постоянно считывать
- 3) На одно прерывание можно повесить хоть 10 энкодеров , главное что-бы количество выводов выбранного МК позволяло
- 4) Работать будет даже «ушатанный» энкодер
- 5) Можно сделать подсветку по вращению энкодера не потребляя при этом ресурсы МК , в представленной ниже схеме при вращении в одну сторону будет мигать красный светодиод , а в другую синий

Минусы :

- 1) Требуется дополнительная обвязка

Итак для начала схема :



Поскольку достойной модели инкрементального энкодера в протеусе не нашлось была применена модель механизм действия которой подробно описан здесь

<http://diymicro.ru/sozдание-ierarxicheskoy-sxemy-v-proteussxematicheskaya-model-enkodera.html>

А здесь ее можно скачать

[https://bitbucket.org/embedgs/diymicro.ru/src/7d439d5cd7382984c4547e37d9e6e0ce0d2b47ce/PicLa b/E20p1\\_Encoder/?at=master](https://bitbucket.org/embedgs/diymicro.ru/src/7d439d5cd7382984c4547e37d9e6e0ce0d2b47ce/PicLa b/E20p1_Encoder/?at=master)

Эта модель по своему принципу действия аналогична тому если-бы мы взяли обычный энкодер и на общий вывод подали-бы лог единицу. С энкодером вроде-бы разобрались, едем дальше.

Прежде чем объяснять принцип действия конвертора вкратце опишу принцип действия D-триггера для тех кто этого не знает. Любой триггер подобен своего рода выключателю , соответственно он может быть установлен (вкл.) или сброшен (выкл.). Когда триггер сброшен на его «прямом» выходе будет лог 0 , а на инвертирующем выходе лог 1 , если-же триггер установиться то на прямом будет лог 1 , а на инвертирующем лог 0.

Что касается специфики именно D-триггера так это просто усложненная модель RS-триггера к которой добавили еще один способ управления, поэтому давайте для начала рассмотрим RS управление.

Следует помнить что управление может быть как «прямым» , тоест осуществляться при помощи лог 1 , так и инвертированным когда управление осуществляется при помощи лог 0 (в нашем случае выводы 1, 4, 10 и 13 инвертированные).

Далее все предельно просто, если подать управляющий импульс на пин «S» то триггер установиться , при этом дальнейшее воздействие на вывод «S» не будет иметь никакого результата. С пином «R» та-же история только триггер сброситься.

Наконец-то мы добрались до второго способа управления ,который осуществляется по пинам «D» и «CLK» , здесь тоже все примитивно - пока нет тактирующих импульсов на пине «CLK» управление триггером через пин «D» невозможно , а если подать тактирующие импульсы то на фронтах этих импульсов триггер начнет реагировать на то что «прилетает» на пин «D» , поскольку конкретно в нашем случае пины «D» и «CLK» НЕ инвертирующие то реагировать мы будем по положительному фронту на «CLK» , и если в этот момент на пине «D» была лог 1 то триггер установиться , а если 0 то сбросится (для инвертирующих пинов реакция идет полностью наоборот).

Ну и на последок стоит отметить что RS-управление имеет приоритет над D-управлением. Это значит что когда подан управляющий сигнал на «R» или «S» , то что-бы не происходило на «D» и «CLK» на них триггер не отреагирует

Итак с теорией работы D-триггера разобрались теперь разберем принцип работы конвертора , для начала изобразим работу энкодера в виде нулей и единиц

| № этапа обработки | 1 | 2 | 3 | 4 |
|-------------------|---|---|---|---|
| Состояние пина А  | 1 | 1 | 0 | 0 |
| Состояние пина С  | 0 | 1 | 1 | 0 |

этап № 1 - на пине А лог 1 , на пине «С» лог 0.

триггер «U1.А» получает лог 1 на пин «R» тем самым деактивируя управление по «RS» (как мы помним они реагируют только на лог 0)

триггер «U1.А» получает лог единицу на пин «D» но пока на него триггер не реагирует ибо не было нарастающего фронта на пине «CLK»

триггер «U1.А» получает лог 0 на пин «CLK» , ничего не происходит

триггер «U1.B» получает лог 0 на пин «R» управление по «D» и «CLK» невозможно , триггер сброшен по «RS»

триггер «U1.B» получает лог единицу на пин «CLK» но триггер не отреагирует так как активно «RS» управление (даже если не обращать на это внимания у нас так и так на пине «D» лог 0 )

триггер «U1.B» получает лог 0 на пин «D», ничего не происходит

этап № 2 - на пине A лог 1 , на пине «C» лог 1.

триггер «U1.A» не меняет состояние на пинах «R» и «D»

триггер «U1.A» получает лог 1 на пин «CLK» , так как во время нарастающего фронта управление по «RS» было неактивно ,а на пине «D» была лог 1 , триггер «U1.A» устанавливается

триггер «U1.B» получает лог 1 на пин «R» тем самым деактивируя управление по «RS»

триггер «U1.B» не меняет состояние на пине «CLK»

триггер «U1.B» получает лог 1 на пин «D»,но ничего не происходит так как нарастающий фронт от единицы что сейчас на пине «CLK» прошел еще на первом этапе

этап № 3 - на пине A лог 0 , на пине «C» лог 1.

триггер «U1.A» получает лог 0 на пин «R» тем самым сбрасывая триггер

триггер «U1.A» получает лог 0 на пин «D» триггер не реагирует из-за активного управления по «RS»

триггер «U1.A» не меняет состояние пине «CLK» , ничего не происходит

триггер «U1.B» получает лог 0 на пин «CLK», ничего не происходит

триггер «U1.B» не меняет состояние на пинах «R» и «D»

этап № 3 - на пине A лог 0 , на пине «C» лог 0.

триггеры «U1.A» и «U1.B» получают на все входы лог 0 , и так как они оба сброшены по «RS» , рассматривать остальные пины нет смысла (желающим никто не запрещает).

Как видно из описания сработает только триггер «U1.A» , если-же вращать энкодер в другую сторону тогда сработает только «U1.B»

Вернемся к нашей схеме , на каждый неинвертирующий вывод поставлен переключающий диод (ультрабыстрый) и после диодов все сведено в 1 линию (подключена к осциллографу на вход D ), именно ее следует пускать микроконтроллеру на вывод с внешним прерыванием настроенным на нарастающий фронт таким образом алгоритм получится следующий :

- 1) Ловим внешнее прерывание с диодов

- 2) В обработчике этого прерывания отключаем прерывания глобально
- 3) По очереди опрашиваем те пины МК куда заведены выходы триггеров на предмет лог 1 , тем самым определяем куда крутили энкодер (на схеме эти пины заведены на А и В входы осциллографа)
- 4) Определяемся какую задачу надо выполнить (тут полная творческая свобода)
- 5) В цикле while ждем пока не пропадет лог 1 с пина куда заведено прерывание (пропадет там == пропадет и на остальных заюзанных пинах)
- 6) Подстрахуемся от того что за время проведенное в обработчике произошло второе прерывание сбросив флаг прерывания (у авр-ок это INTFx)
- 7) Разрешим прерывания глобально

Непосредственно сам код приводить не буду ибо на схеме нет МК , но полагаю по приведенной выше инструкции минимальных знаний ЯП хватит что-бы юзать такую конструкцию с любым типом микроконтроллеров.

Теперь, когда мы разобрались с основным функционалом перейдем к второстепенному.

Для подсветки действий юзера очень кстати у наших триггеров имеются инвертирующие выходы , хоть по даташиту на них макс нагрузка по 20 миллиампер чего достаточно для подключения светодиодов без транзисторов , лично я так делать настоятельно не рекомендую, и в своих разработках предпочитаю разгружать микросхемы по максимуму

В целях увеличения надежности и помехозащищенности «S» пины каждого из триггеров подтянуты к плюсу питания резюком 1 КОм , это нужно для того что-бы исключить деятельность этого вывода от помех , так-же не забываем про резисторы R6 – R8 , возможно конечно встроенной в МК подтяжки и хватит, но симуляция в протеусе показала что без них могут быть глюки.

(с) асскyіbоxxx

Файлы к статье см. в аттаче.