

Микроконтроллеры STM32

Домашнее задание №9

Евгений Зеленин

31 марта 2025 г.

1. Постановка задачи

Условия задачи.

Перед вами простейшая схема позволяющая с помощью симистора управлять мощной нагрузкой. Схема распространена на сайтах сторонников проекта Arduino.

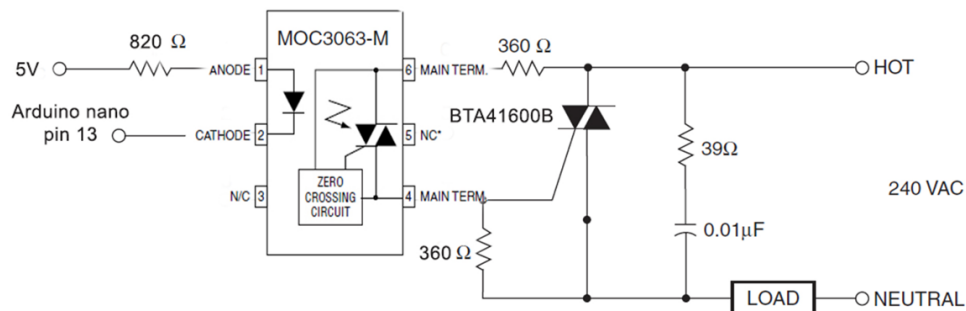


Рис. 1: Подключение Arduino

Модифицируйте данную схему таким образом, чтобы мощность в нагрузке плавно изменялась под управлением микроконтроллера (на выбор учащегося). Целью домашнего задания является изучение подключения внешних цепей к микроконтроллерам, повторение ранее изученного на курсе “основы электроники” материала. Результатом выполнения данного домашнего задания должна быть схема с комментариями содержащая микроконтроллер и необходимые цепи обвязки. Схема и пояснения предоставляются в формате PDF.

2. Доработка схемы

По заданию, схему требуется модифицировать для обеспечения возможности плавной регулировки мощности. Как известно, принцип действия симистора заключается в следующем: он управляет нагрузкой в цепи переменного тока и открывается под воздействием управляющего сигнала, а закрывается в момент когда снимается напряжение (переход через ноль) или ток удержания опускается ниже порогового. Таким образом, чтобы точно регулировать мощность нужно синхронизировать управляющий сигнал с переходом питающего напряжения через ноль. Для этого, соберем схему детектирующую переход через ноль и подающую импульс на вход МК через гальваническую развязку. На рисунке 2 показана схема Cross Detection. В момент перехода через ноль на выходе схемы формируется импульс длительностью примерно в 1мс. Следует отметить, что передний фронт импульса опережает момент пересечения примерно на 0.8с по результатам симуляций, это требуется учесть при написании алгоритма регулирования.

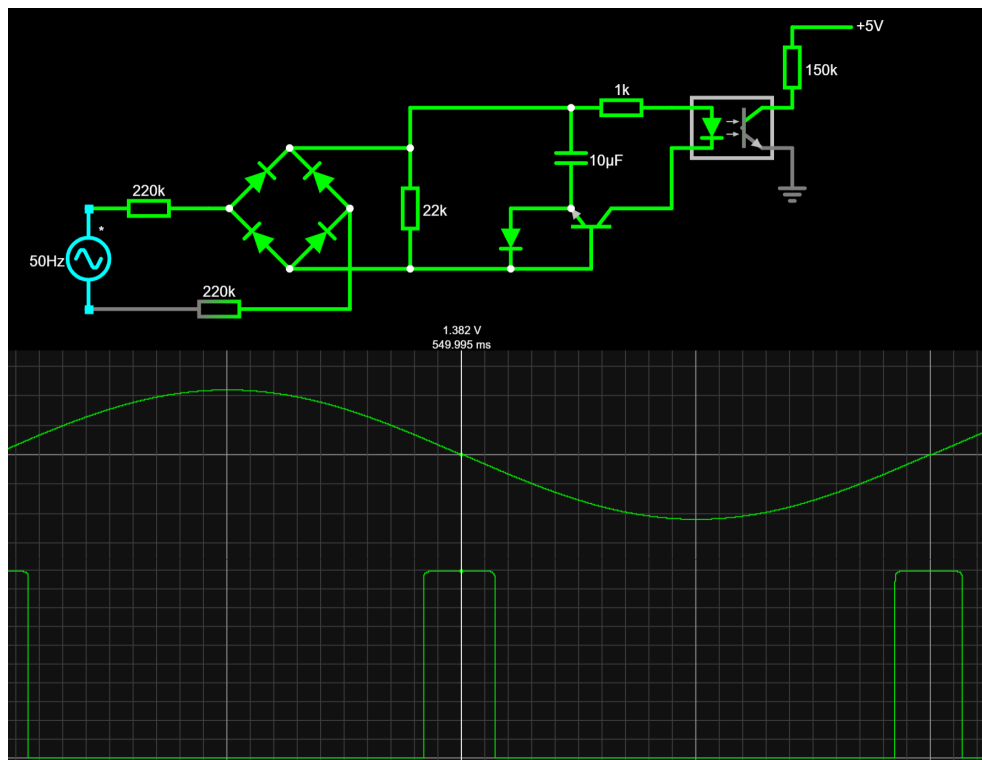


Рис. 2: Симуляция схемы Cross Detection

Схема работает следующим образом (Рисунок 3). В момент накопления энергии (когда на выходе моста присутствует напряжение), конденсатор C4 заряжается через диод D5. Транзистор Q2 в это время закрыт, ток через оптопару U2 не течет. Когда напряжение на выходе моста становится ниже чем напряжение на конденсаторе C4 и падение на переходе база-эмиттер транзистора Q2 ($U_{C4} - U_{be}$), заряд конденсатора прекращается и он начинает отдавать накопленную энергию. Транзистор Q2 открывается через резистор R8 (контур R8 - Q2 - C4), после чего через оптопару U2 начинает протекать ток по цепи C4-R9-Q2, оптопара срабатывает, формируется передний фронт импульса синхронизации. Когда напряжение на выходе моста снова превышает напряжение на конденсаторе C4, он прекращает отдавать заряд, транзистор Q2 закрывается, ток через оптопару не течет, формируется задний фронт импульса синхронизации.

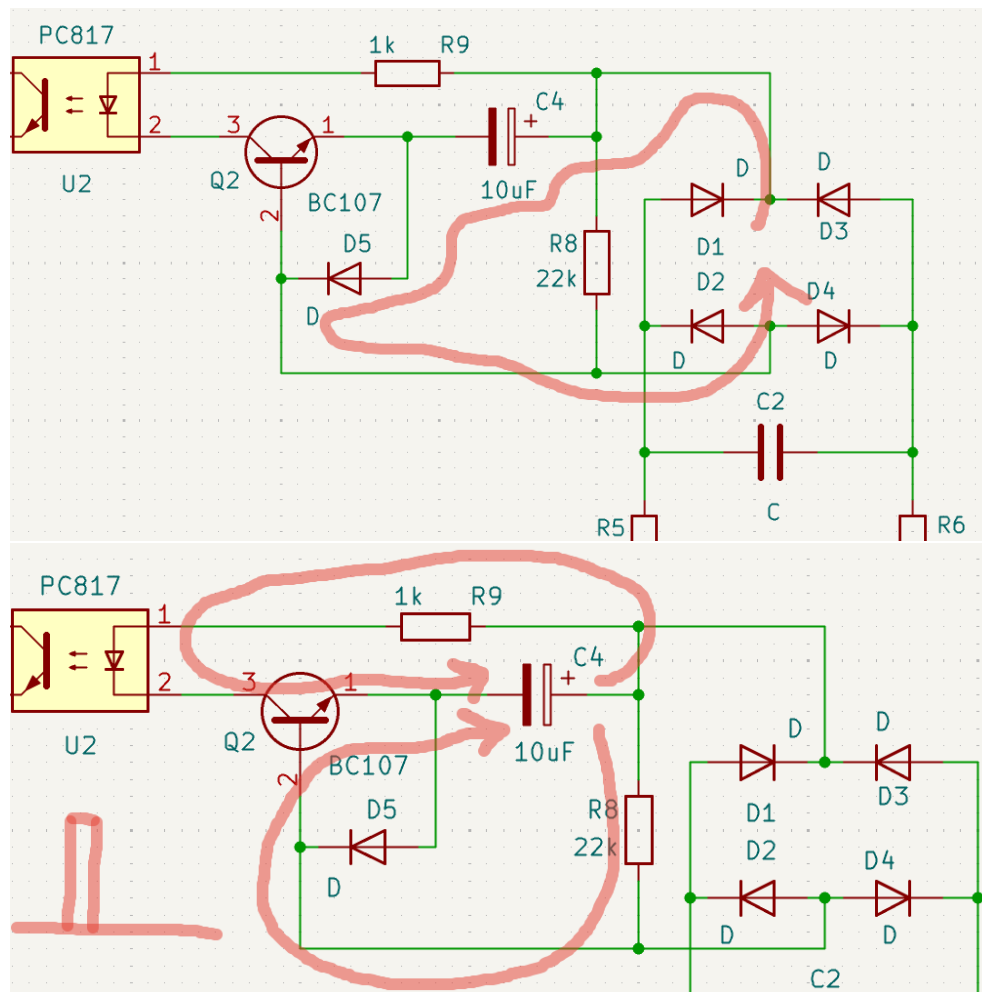


Рис. 3: Принцип действия

Объединим модуль определяющий пересечение ноля с модулем, управляющим нагрузкой. На рисунке 4 показана схема силовой части будущего устройства, остается лишь выбрать подходящий МК и нарисовать цепи обвязки.

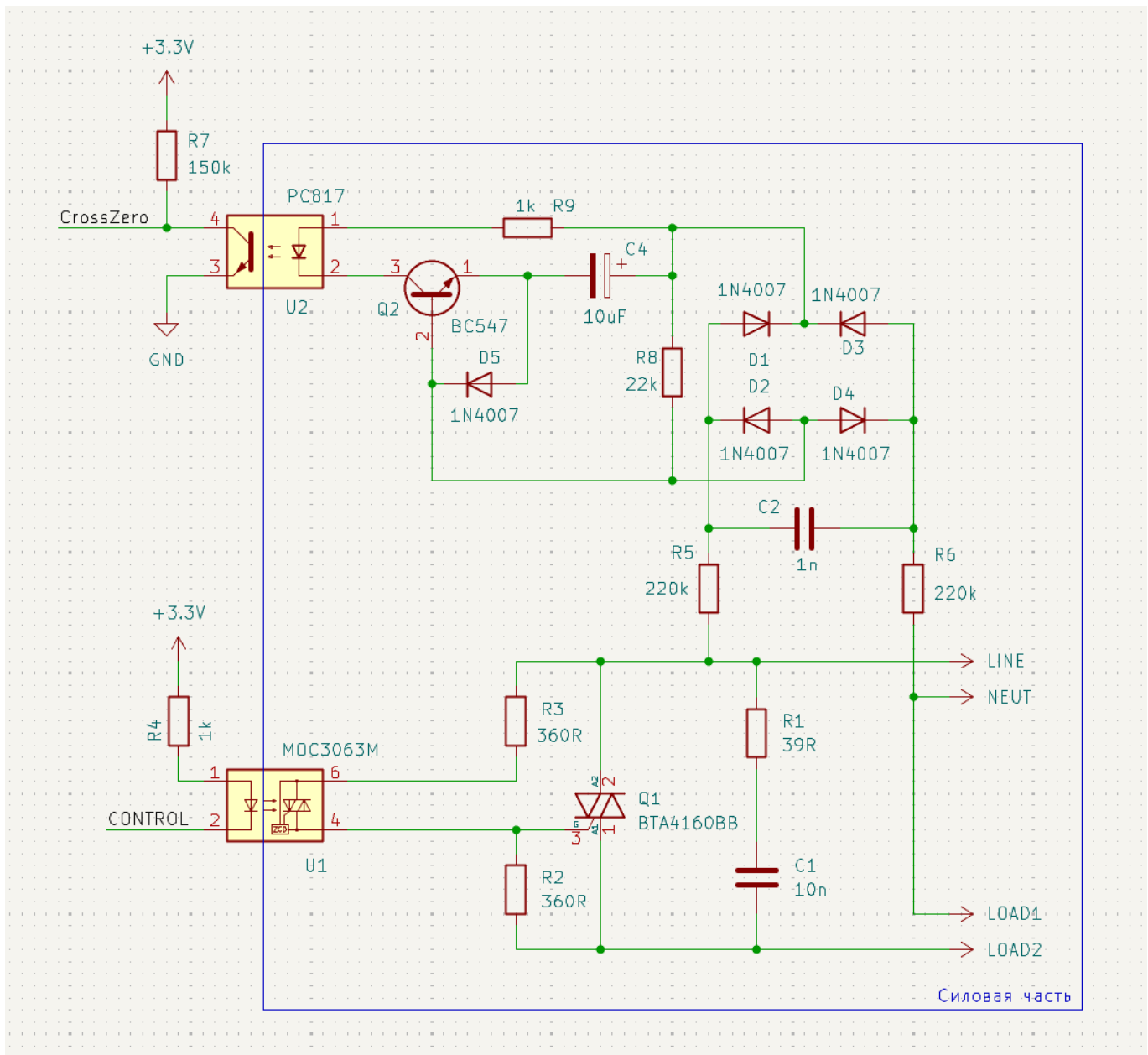


Рис. 4: Силовая часть устройства

3. Выбор МК и составление схемы обвязки

Для управления нагрузкой и получения обратной связи более чем достаточно двух портов на микроконтроллере. Причем, управление можно сделать даже без участия процессора МК, если со схемы CrossZero управляющий сигнал использовать в качестве триггера таймера, а таймер настроить таким образом, чтобы он выдавал управляющий импульс после отсчета заданного количества тактов. Наверняка, нам потребуется еще какое-либо управление (подключение энкодера, фотодатчика, uart, i2c и т.д.). Ограничимся подключением энкодера и фото-датчиком. Для этого потребуется еще 4 вывода МК. В качестве генератора системной частоты используем встроенный генератор. Еще минимум два вывода требуется для питания устройства. Итого, минимум 8 выводов. Если совместить выходы SWD, NRST, BOOT с GPIO, то для текущего функционала достаточно МК STM32C011J4M в миниатюрном кор-

пусе SO8N.

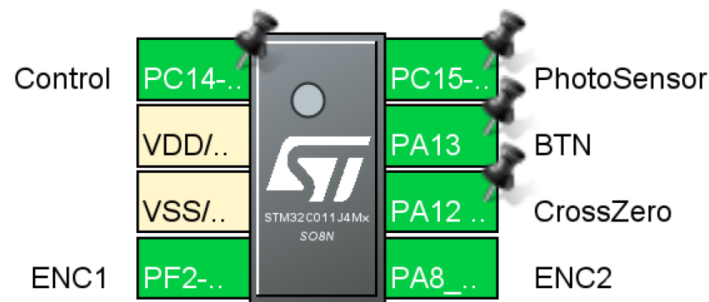


Рис. 5: Назначение выводов МК

Через option bytes можно настроить nrst таким образом, что после включения будет возможно использовать его в качестве порта ввода - вывода. Достаточно лишь обеспечить высокий уровень на nrst в момент включения.

Reset pin (PF2-NRST) in GPIO mode

The PF2-NRST pin can be configured as reset I/O or as a GPIO.

To configure PF2-NRST as a GPIO (input, output, AF, or analog I/O), set the NRST_MODE bitfield to GPIO mode in the FLASH option bytes. The new setting only takes effect upon the option byte loading (OBL) event following a reset. Until the reset release, PF2-NRST keeps acting as reset I/O.

The user must ensure that, upon power-on, the level on the NRST pin can exceed the minimum $V_{IH(NRST)}$ level specified in the device datasheet. Otherwise, the device does not exit the power-on reset. This applies to any NRST configuration set through the NRST_MODE[1:0] bitfield, the GPIO mode inclusive.

When PF2-NRST acts as a GPIO, reset can only be triggered from one of the device internal reset sources and the reset signal cannot be output.

For further information on reset function, refer to the *RCC* section.

Рис. 6: Переназначение nrst

Аналогично, boot0 после отладки можно настроить через option bytes и совместить его с GPIO, а далее предусмотреть программную обработку.

Отладочный интерфейс SWD также совмещен с другими портами и отключается после прошивки/отладки устройства. Для отделения цепей в момент отладки предусмотрены перемычки JP1 и JP2 (Рисунок 8).

Bits 28:27 NRST_MODE[1:0]: PF2-NRST pin configuration
00: Reserved
01: Reset input only: a low level on the NRST pin generates system reset; internal RESET is not propagated to the NRST pin.
10: PF2 GPIO: only internal RESET is possible
11: Bidirectional reset: the NRST pin is configured in reset input/output (legacy) mode

Bit 26 NBOOT0: NBOOT0 option bit
0: NBOOT0 = 0
1: NBOOT0 = 1

Bit 25 NBOOT1: NBOOT1 boot configuration
Together with the BOOT0 pin or NBOOT0 option bit (depending on NBOOT_SEL option bit configuration), this bit selects boot mode from the main flash memory, SRAM, or the system memory. Refer to [Section 3: Boot modes](#).

Bit 24 NBOOT_SEL: BOOT0 signal source selection
This bit defines the source of the BOOT0 signal.
0: BOOT0 pin (legacy mode)
1: NBOOT0 option bit

Рис. 7: Option bytes

Объединим все вышесказанное и составим схему будущего устройства. Как видно из рисунка 8, кнопка сброса совмещена с выводом энкодера ENC1 и в момент включения на нем присутствует высокий уровень напряжения.

Boot0 совмещен с фото-датчиком (Канал 14 АЦП). Конденсатор С6 требуется для того, чтобы в момент первого включения обеспечить высокий уровень и выполнить запуск с встроенного загрузчика. После установки options bytes логика работы Boot0 меняется и вместо загрузчика запускается основное ПО. Также, возможно полностью отключить аппаратный Boot0 и обрабатывать его программно в коде.

Энкодер подключен с использованием TVS-диодов для защиты от статического электричества.

Вывод Control служит для управления оптопарой и предполагается его настройка по схеме OpenDrain.

Вход CrossZero может служить как источником триггера для таймера, так и источником прерывания или события и далее обрабатываться программно.

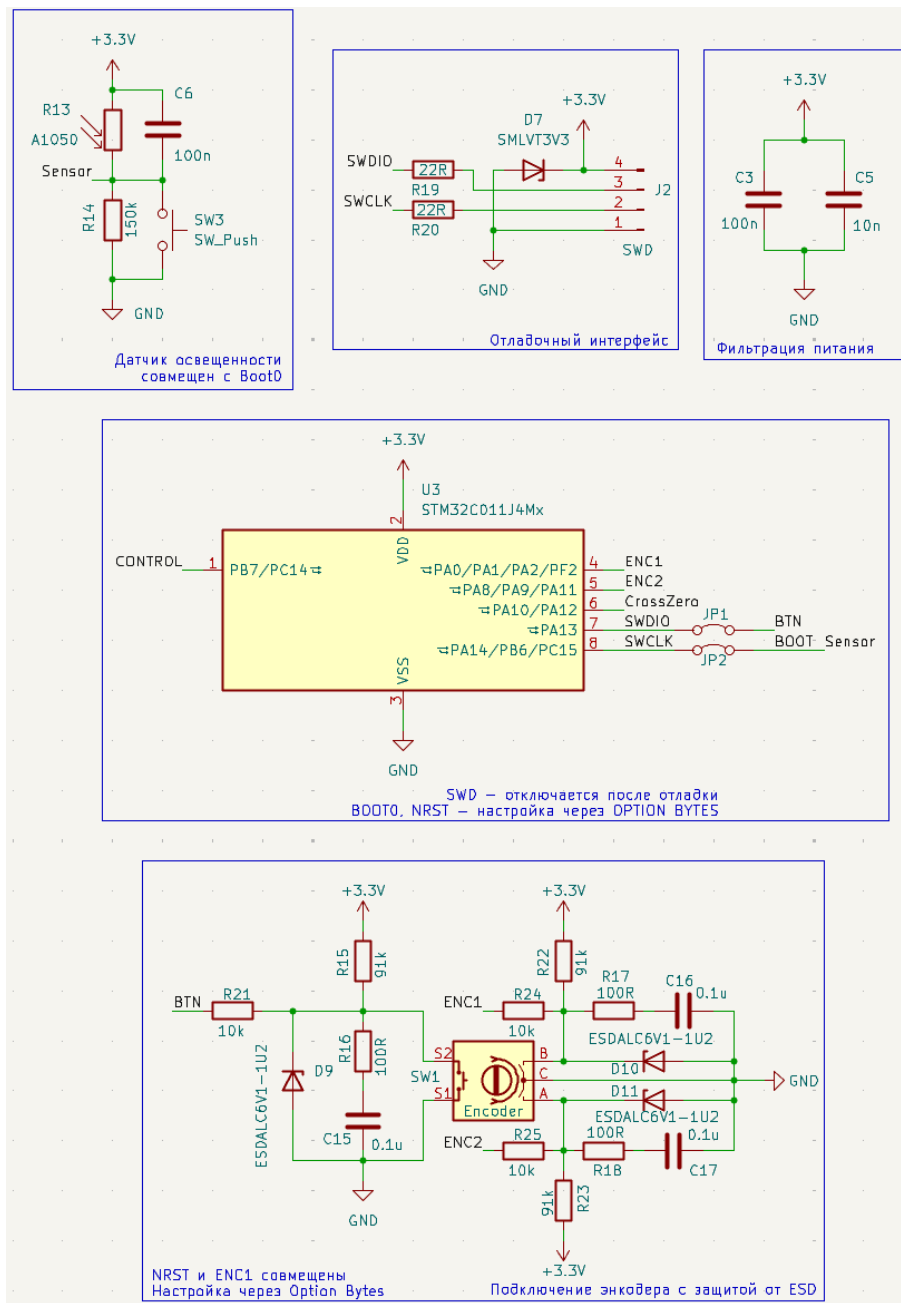


Рис. 8: Обвязка МК

4. Логика работы

Опишем предполагаемый режим работы устройства. Для управления мощностью, будем использовать информацию о моменте перехода через ноль питающего напряжения. Устанавливая задержку между переходом через ноль и подачей управляющего импульса на симистор, можно управлять временем в течение которого каждый полупериод через нагрузку не протекает ток. Соответственно, мощность выделяемая на нагрузке будет зависеть от соотношения T_h/T . Где T_h - период, в течение которого симистор открыт, а T - длительность всего полупериода (10мс)

5. **Дополнительные материалы**

Материалы к отчету расположены в папке на google диск по следующей ссылке: Материалы к ДЗ №09