

# Введение в микроконтроллеры

Курсовой проект

Евгений Зеленин

21 февраля 2025 г.

## 1. Постановка задачи

### Условия задачи.

Сделайте стабилизированный источник регулируемого напряжения. Для настройки можете использовать потенциометр, энкодер, кнопки или данные от другой платы (скажем, ардуино, выполняющей роль интерфейса — через UART или SPI). Уставку и обратную связь по напряжению выводите на экран, подключённый к stm или ардуино. \*Добавьте шунт в цепи нагрузки (один или несколько параллельно резисторов на 1 Ом — в зависимости от требуемого тока) и выведите потребляемый ток. \*\*Выход предусмотрите двухтактный при помощи транзисторов разной проводимости (если вы хотите питать полумост от напряжения больше 3,3 вольта, позаботьтесь о защите входа АЦП: резистивный делитель + стабилитрон; также рекомендую защитить транзисторы обратными диодами). В качестве отчёта принимается схема устройства + код + иллюстрации / короткое видео и / или текстовое описание работы.

## 2. Описание проекта

По заданию, требуется сделать источник стабилизированного регулируемого напряжения. Для настройки уставки будем использовать энкодер. Также, к контроллеру подключим SPI дисплей ST7739 и выведем на него значение уставки, текущего выходного напряжения, тока нагрузки и коэффициент заполнения ШИМ. Для измерения тока будет использоваться шунт 0.5 Ом (два сопротивления по 1 Ом параллельно). Силовую же часть устройства соберем по схеме полумоста с возможностью использовать напряжение питания больше 3.3в.

## 3. Разработка схемы

Для выполнения этой работы установим следующее ограничение: будем использовать доступные компоненты из набора по электронике и рекомендованную слушателям курса отладочную плату mcsuinside. Подключение таких элементов как экран и энкодер типовое и не вызывает каких-либо дополнительных вопросов, поэтому сосредоточим внимание на силовой части.

Суть проекта - в демонстрации работы регулятора с использованием двух каналов шим (PWM и PWMN) для управления работой полумоста. Поэтому, чтобы не усложнять конструкцию, ограничимся использованием npn и pnp транзисторов bc337 и bc327. Максимально допустимый ток которых достигает 0.5а, а рассеиваемая мощность - 625мВт, чтобы не выйти за допустимые пределы по мощности, ограничим ток выходного каскада ( $Q_1, Q_3$ ) с помощью двух включенных параллельно резисторов ( $R_9, R_{10}$ ) на 10 Ом. На выходе регулятора будем использовать сглаживающий конденсатор на 470мкф.

Питание силовой части будет осуществляться от 5в, поэтому, для получения обратной связи по напряжению требуется использовать резистивный делитель 1:2. Для получения обратной связи по току будем использовать шунт из двух резисторов по 1 Ом (0.5 Ом) и защитный резистор ( $R_{14}$ ) на 91к.

Для согласования уровней управляющих сигналов используются транзисторы  $Q_2, Q_4$ .

Ограничением такой реализации (рисунок 1) является невозможность регулирования напряжений ниже 0.45в без нагрузки. Это связано с тем, что для открытия  $Q_3$  требуется разность потенциалов между базой и эмиттером не менее 0.7в. Соответственно, минимально возможный уровень выходного напряжения будет примерно равен  $V_{min} = V_{be} - U_{R_{10}}$ . По результатам моделирования (рис 2), эта величина составила около 0.6-0.7в. Путем измерений на практике установлено, что этот уровень соответствует 0.45в. При более низком уровне напряжения, величина тока разряда конденсатора настолько мала, что скорость разряда становится крайне низкой и возможность регулирования нарушается.

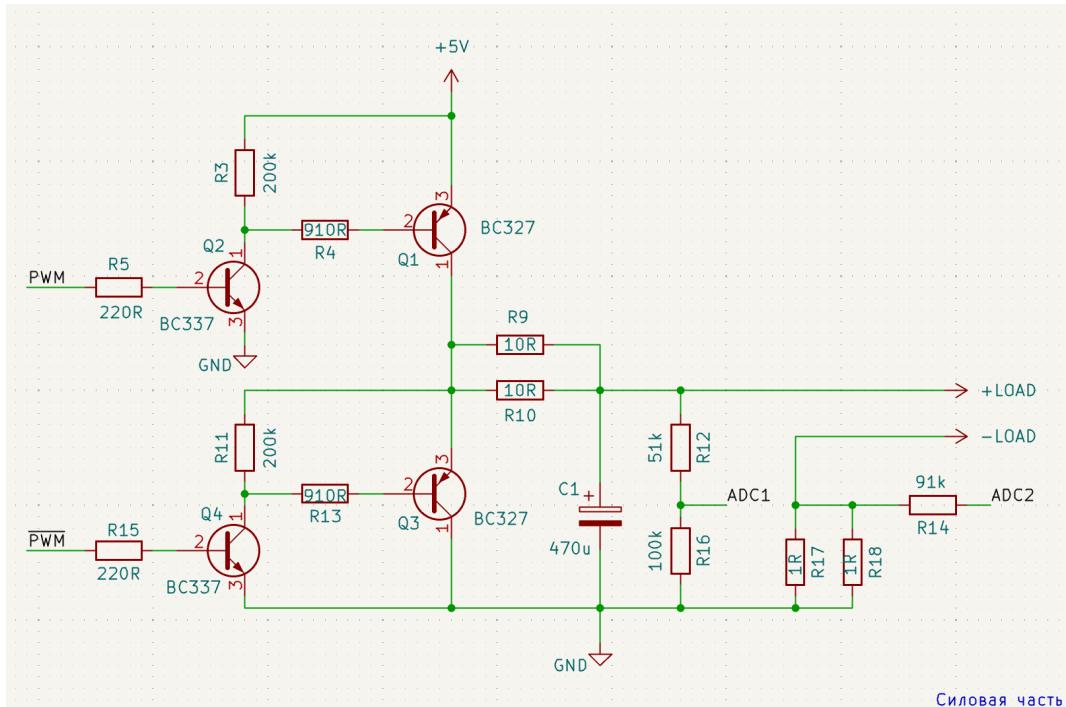


Рис. 1: Схема силовой части

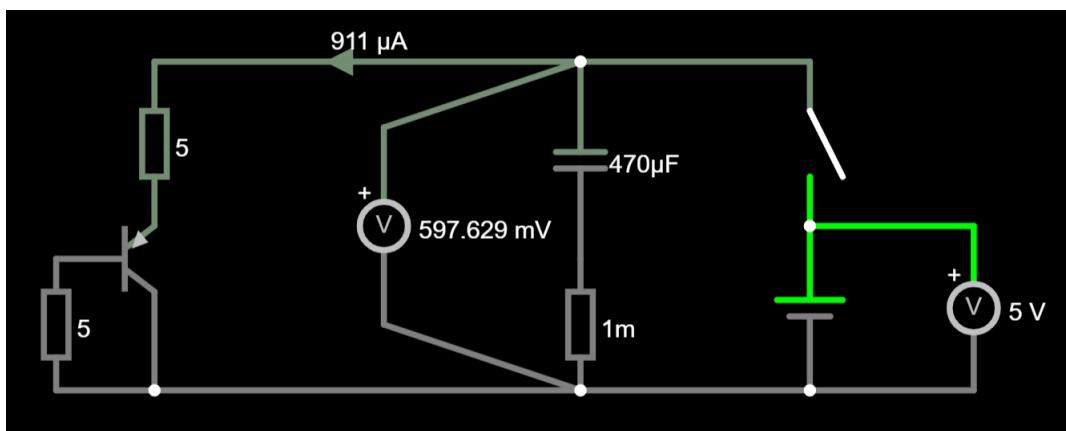


Рис. 2: Моделирование минимального нижнего уровня напряжения

На рисунке 3 показаны подключения к мк.

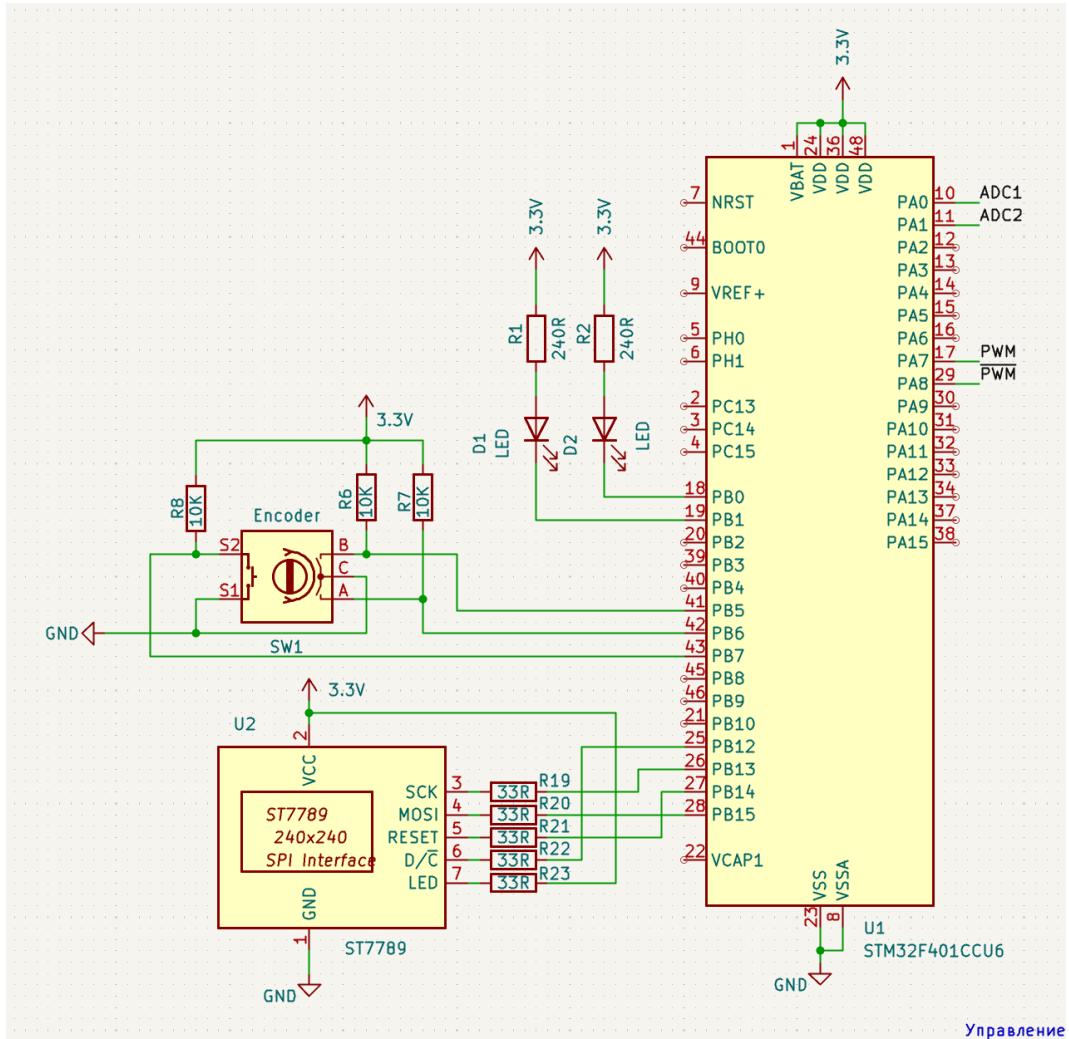


Рис. 3: Подключения к МК

#### 4. Сборка силовой части

Силовая часть была собрана на монтажной плате (рисунки 4, 5).

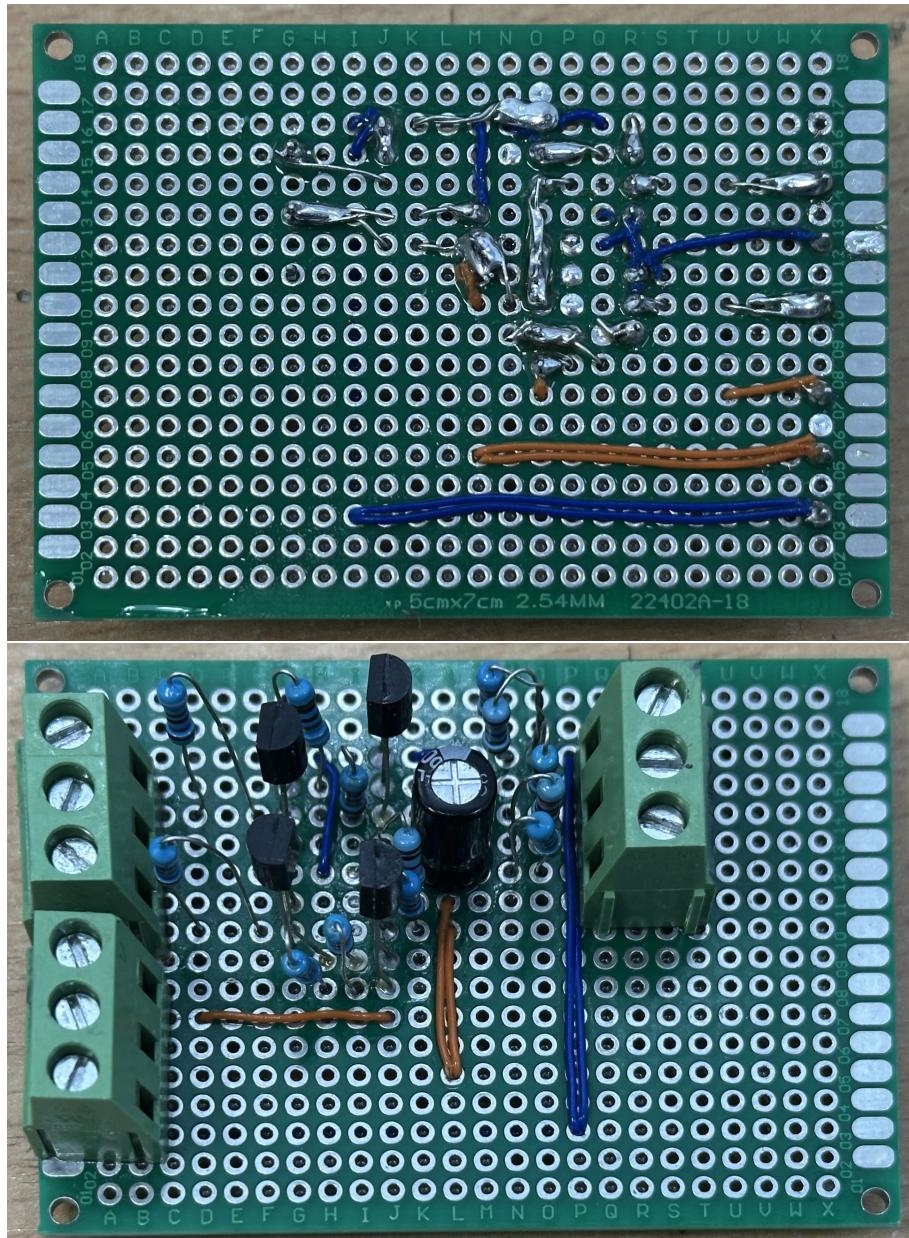


Рис. 4: Сборка силовой части

## 5. Работа ШИМ

В нашем случае, использован двухканальный режим работы ШИМ. Задержка между переключениями каналов 0.5мкс. Ниже показано несколько осциллографм отражающих основные моменты работы ШИМ (рисунки 5, 6, 7):

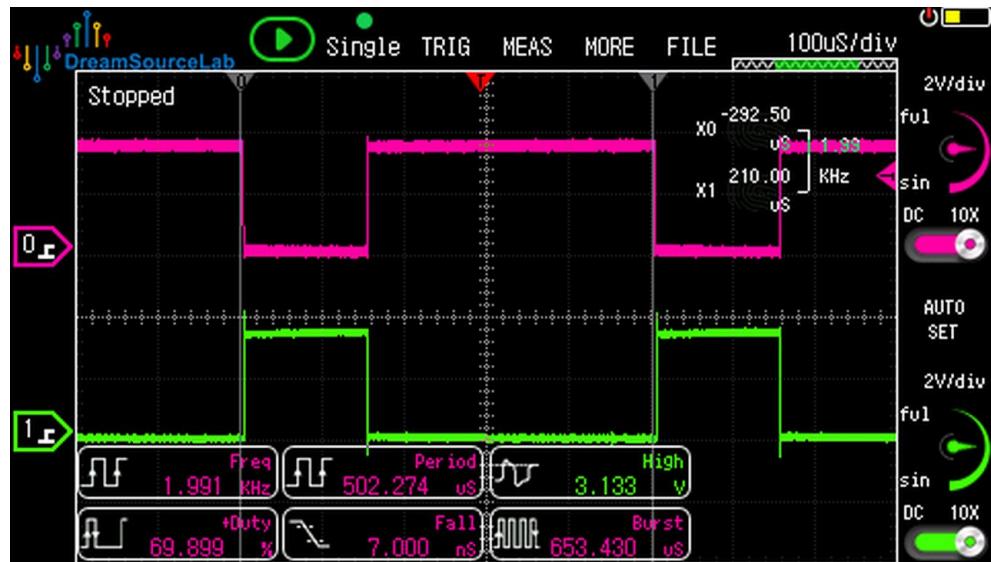


Рис. 5: Парная работа каналов ШИМ

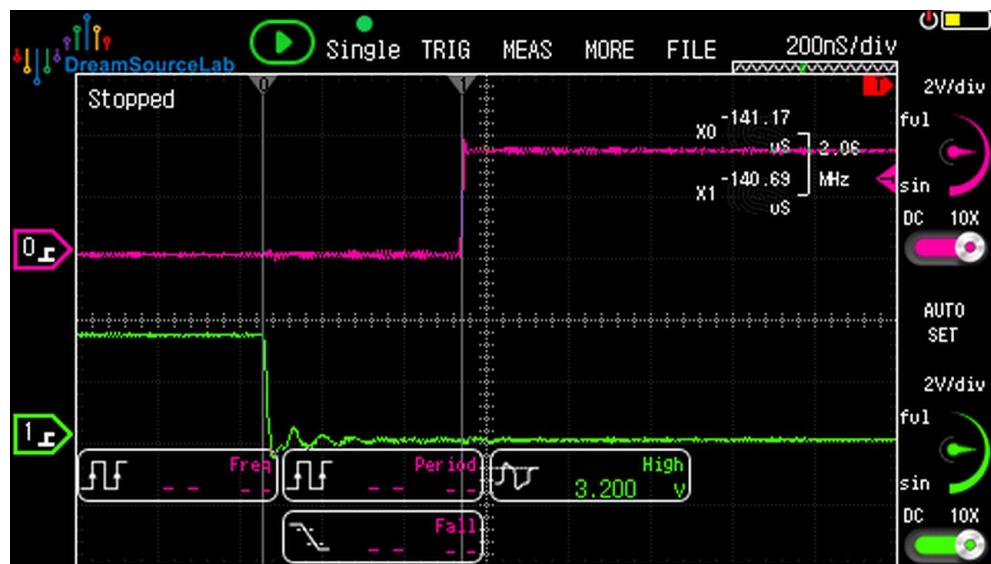


Рис. 6: Задержка между переключением каналов

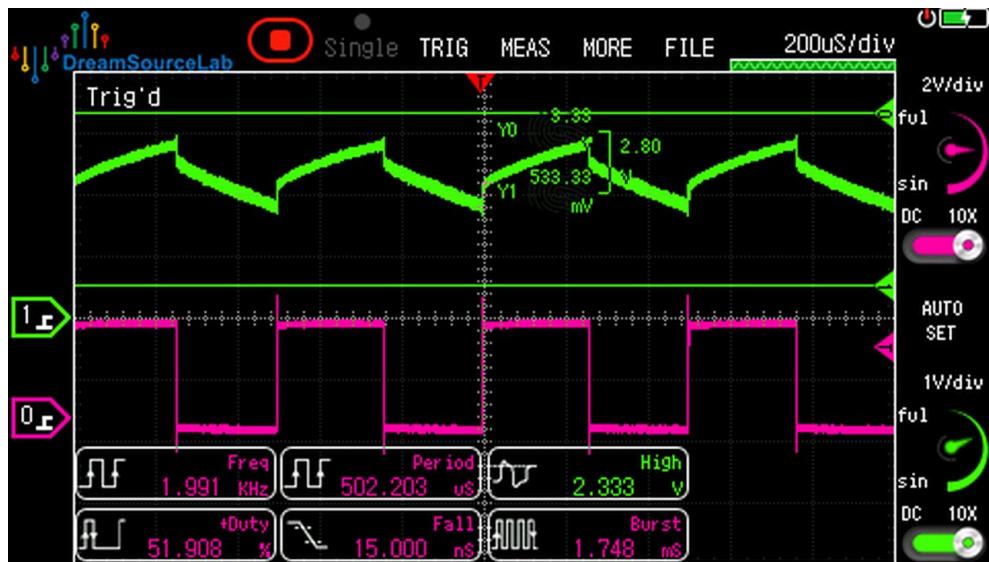


Рис. 7: Канал PWM и выход регулятора без нагрузки

Измерим температуру транзисторов во время регулирования при нагрузке в 100 Ом.

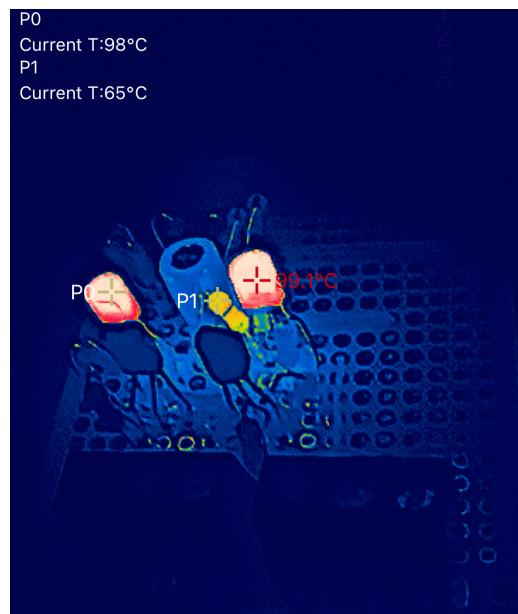


Рис. 8: Измерение температуры транзисторов

На рисунке 8 показана ситуация при коэффициенте заполнения около 45%. В этом случае, температура транзисторов примерно одинакова и составила 100 градусов по цельсию. При уменьшении коэффициента заполнения транзистор нижнего плеча разогревается до 130с, а верхнего остывает до 30-40с. При увеличении коэффициента заполнения ситуация меняется на противоположную.

Что интересно, при подключении микро-мотора вместо резистора, ситуация (в плане максимальных и минимальных температур) практически не меняется.

Максимально допустимая рабочая температура для bc327/337 - 150с, транзисторы работают на пределе своих возможностей.

## 6. Работа каналов АЦП

По заданию, чтобы обеспечить возможность регулирования, требуется измерять уровень выходного напряжения. Для этого используется регулярный канал АЦП №1, преобразование запускается по событию с ТИМ2, сброс которого синхронизирован с передним фронтом ШИМ. Периодичность опроса - 12.5мкс. Для повышения точности измерения используется измерение Vref в инжектированном режиме (автоматический запуск сразу после регулярного преобразования).

Помимо напряжения, нам требуется выводить на дисплей потребляемый ток. Для этого используется канал №2. Преобразование, также, осуществляется в инжектированном режиме.

Синхронизация каналов АЦП с шим показана на рисунке 8:

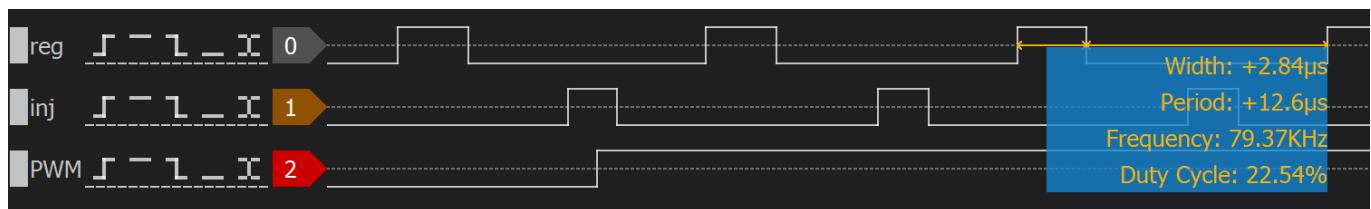


Рис. 9: Синхронизация каналов АЦП

Подключим разные виды нагрузок и посмотрим на осцилограммы токов и напряжений на входах АЦП (рисунки 9, 10). Канал желтого цвета отображает ток (напряжение на шунте), канал бирюзового цвета - напряжение на нагрузке.

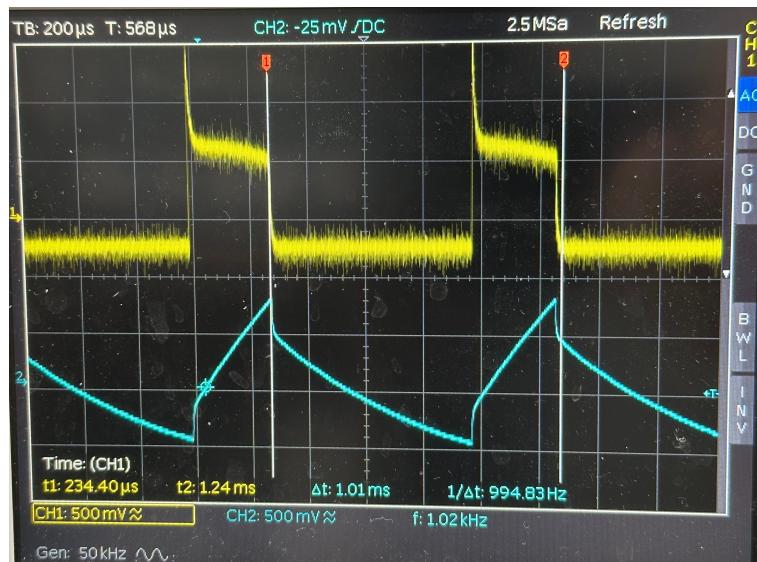


Рис. 10: Подключена только емкостная нагрузка (конденсатор)

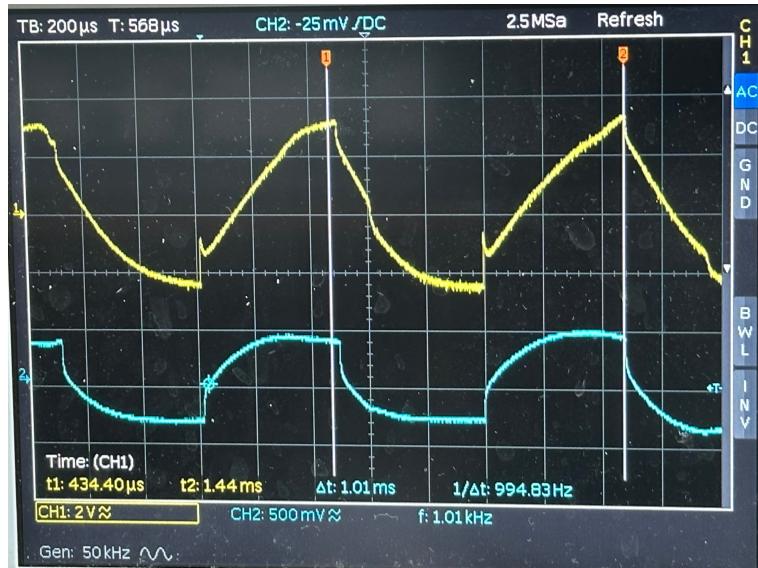


Рис. 11: Добавлена индуктивная нагрузка (мотор)

Как видно из рисунков, характер поведения емкостной и индуктивной нагрузки кардинально отличается. При подключении в качестве нагрузки конденсатора, во время фазы заряда наблюдается резкий всплеск тока, по мере заряда токовая составляющая снижается, а уровень напряжения растет. В момент переключения полумоста на разряд, ток через нагрузку исчезает, а напряжение на нагрузке начинает уменьшаться.

В случае же подключения микро-мотора, ситуация иная - индуктивность сопротивляется как заряду, так и разряду, поэтому после переключения полумоста на разряд, ток нагрузки не исчезает мгновенно: индуктивность начинает бороться с конденсатором, постепенно разряжаясь. В случае "победы" будем наблюдать отрицательный выброс на осциллографе. В таких случаях, нужно предусмотреть защиту от выбросов, хотя бы с помощью диода. Но мы такую мощную нагрузку в этой работе подключать не будем, поэтому хватит и сопротивления в 91к и встроенных в мк защитных диодов.

Стоит отметить, что в целях регулирования и отображения будет использоваться усредненное значение, полученное с помощью цифровой фильтрации полученных измерений.

На рисунке 11 показан регулятор в процессе работы с подключенным моторчиком. Как видно из рисунка, ток, отображаемый на SPI дисплее, достаточно близко совпадает со значением измеренным с помощью мультиметра.

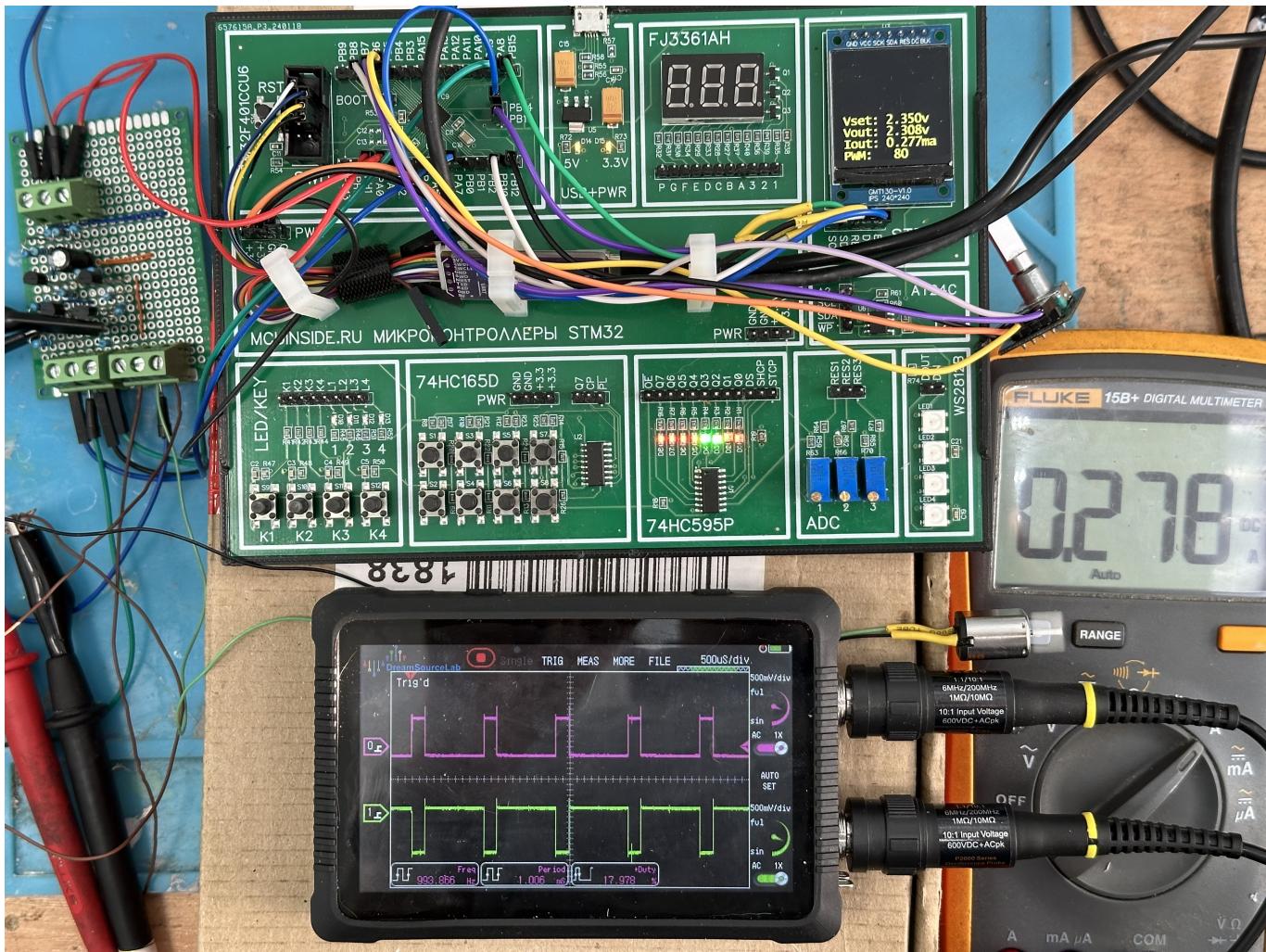


Рис. 12: Регулятор в процессе работы

## 7. SPI дисплей

Дисплей подключен к порту SPI2 на частоте 21МГц длинными проводами. Не смотря на наличие помех и искажений в линиях связи, проблем в работе не было замечено (рисунок 13). Демонстрация этого момента отражена на видео в материалах к занятию.

Что интересно, при подключении к SPI1 на этих же 21МГц наблюдались проблемы со связью дисплея и мк, связь терялась даже от простого прикосновения к проводам.

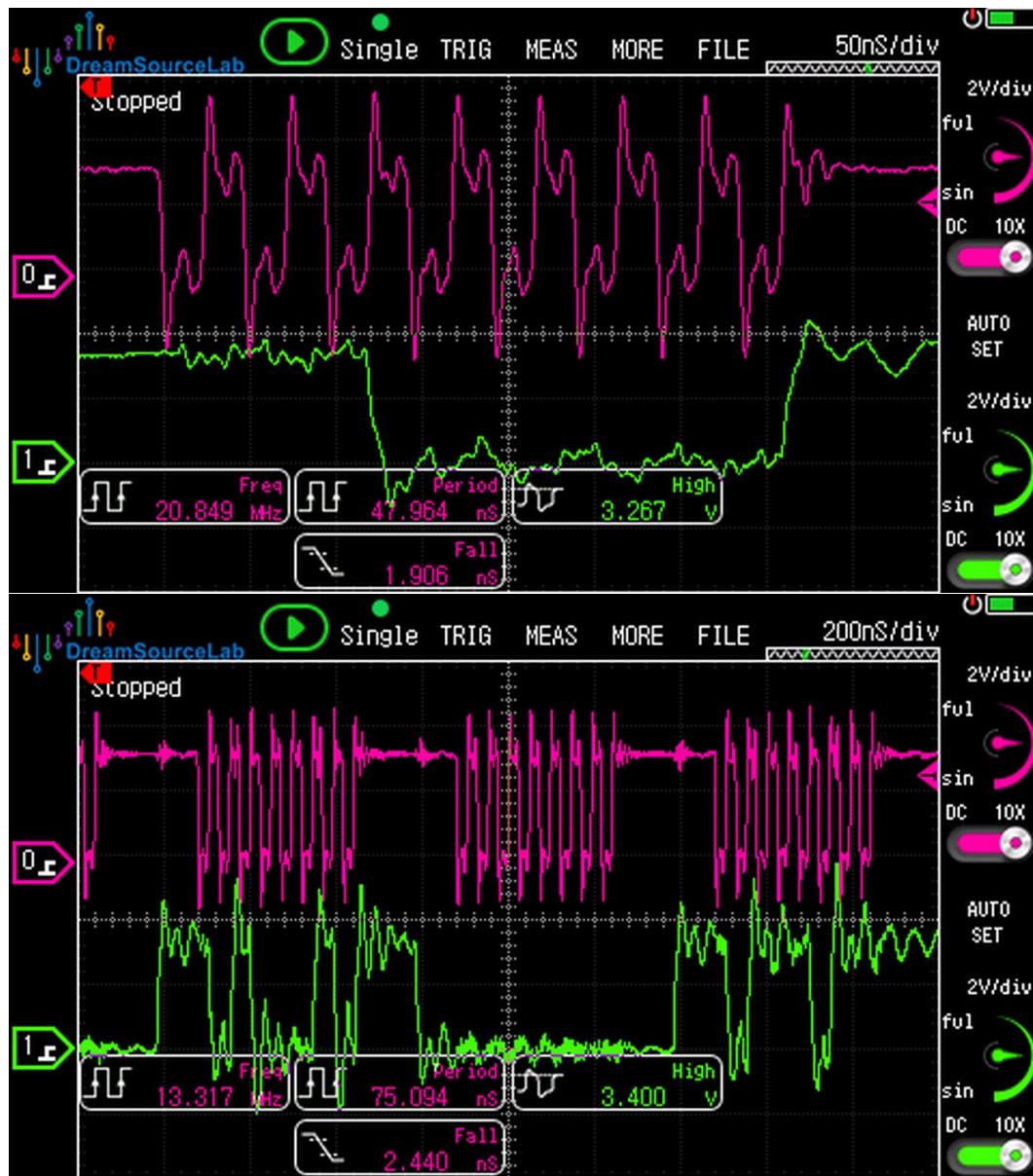


Рис. 13: Осциллографы линий SCK и MOSI на 21МГц

После добавления в каждую линию связи сопротивления 33 Ом. Проблемы со стабильностью соединения исчезли и дисплей заработал на частоте 42МГц на SPI1.

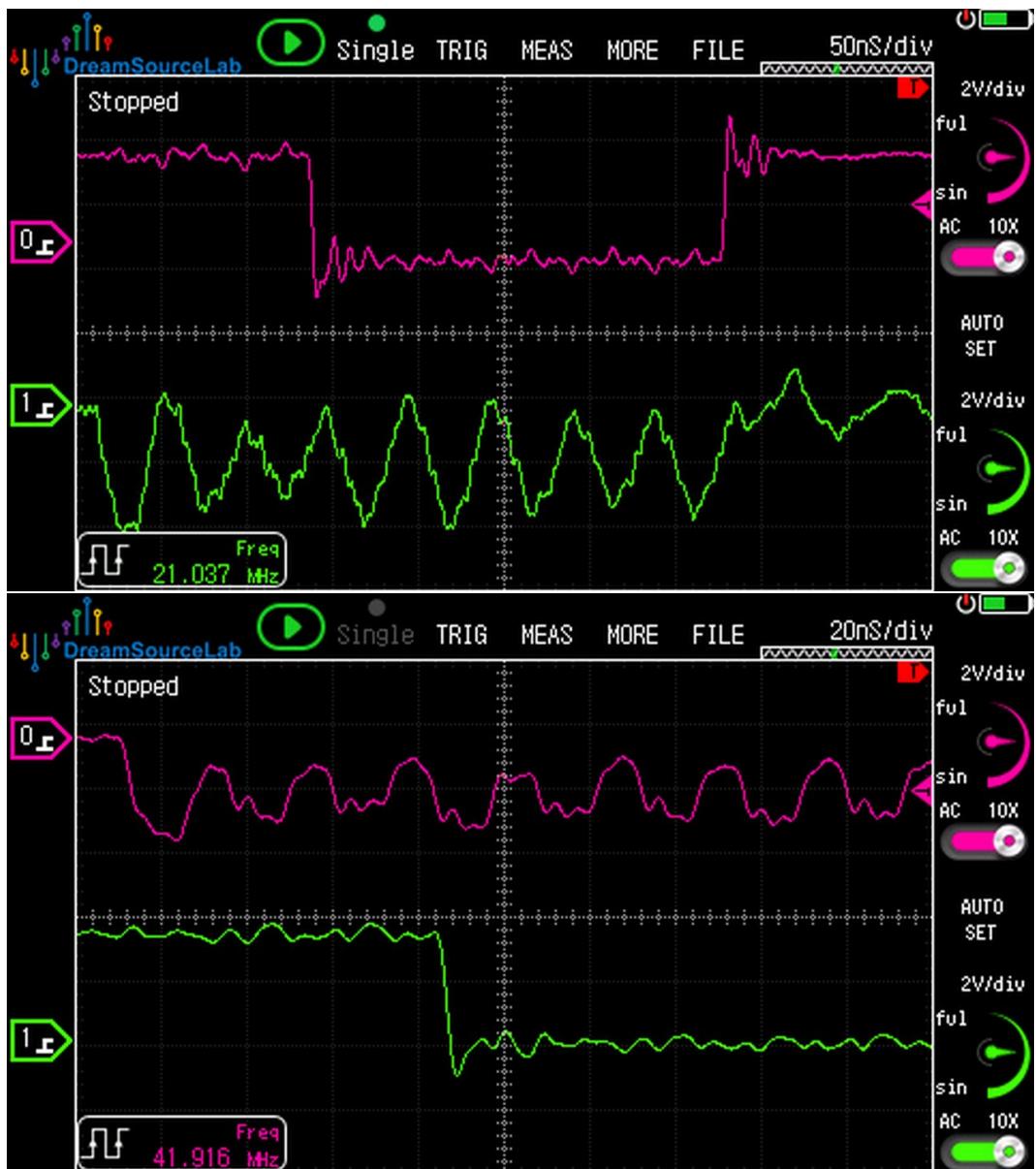


Рис. 14: Осциллографмы линий SCK и MOSI на 21МГц и 42МГц с резистором 33 Ом

## 8. Программная реализация

Программная часть устройства разделена на несколько модулей:

- my\_pid - модуль, отвечающий за работу PID регулятора
- my\_psu - инициализация и работа с пользовательской периферией, интерфейс
- my\_task - реализация простого диспетчера задач
- st7789 - библиотека для работы с дисплеем

После включения устройства, пользователь с помощью энкодера устанавливает желаемое выходное напряжение. Задача PID регулятора - поддерживать эту величину с помощью изменения коэффициента заполнения ШИМ.

Чтобы реализовать многозадачность используется диспетчер задач и прерывания. Всего в проекте созданы две задачи: display() - отображает данные на дисплее и run\_PID() - запускает PID регулятор с заданной периодичностью.

В прерываниях же обрабатывается такие события как вращение энкодера и преобразование АЦП. Ниже приведены наиболее существенные фрагменты исходного кода из различных модулей. Полный код проекта приложен в материалах к занятию.

### Обработка результатов АЦП преобразования

Чтобы работа PID регулятора была более стабильной, требуется фильтрация входных данных с АЦП преобразователя. Для этого будем использовать целочисленную реализацию цифрового фильтра "бегущее среднее" со следующими коэффициентами  $f_k = 6$ ,  $f_A = 32$ ,  $f_B = 32$ .

```
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc) {
    HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, GPIO_PIN_SET);
    if (hadc == &hadc1) {
        adcv = (fA * adcv + fB * HAL_ADC_GetValue(&hadc1)) >> fK;
    }

    HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, GPIO_PIN_RESET);
}

void HAL_ADCEx_InjectedConvCpltCallback(ADC_HandleTypeDef *hadc) {
    HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, GPIO_PIN_SET);
    if (hadc == &hadc1) {
        adci = (fA * adci
            + fB * HAL_ADCEx_InjectedGetValue(&hadc1, ADC_INJECTED_RANK_1))
            >> fK;
        vref = (fA * vref
            + fB * HAL_ADCEx_InjectedGetValue(&hadc1, ADC_INJECTED_RANK_2))
            >> fK;
    }

    HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, GPIO_PIN_RESET);
}
```

## Вывод информации на дисплей

Результаты измерений АЦП не соответствуют реальным значениям, так как для измерения используются резистивные делители. Поэтому, полученные данные требуется пересчитать в требуемый диапазон в соответствии с коэффициентом делителя. Для повышения точности измерений, также следует учесть значение опорного напряжения vref.

```
void display(void) {
/*
 * vref = 1200mv
 * 151k/100k - resistive divider
 * vext - defined value of external voltage supply
 * 3300mv - maximum output value of divider
 * vext /3300 *(1200 * 151/100) = 1812
 * 1805 - adjusted with using external multimeter
 */
vout = (vext/3300) * (adcv * 1805) / vref;

/*
 * there is used 0.5 ohm resistor to measure current
 * 950 coeff was adopted with using external multimeter
*/
iout = (adci * 950) / vref;

sprintf(buffer, "Vset: %lu.%03lu", vset_mv / 1000, vset_mv % 1000);
ST7789_WriteString(10, 240 - 4 * 26, buffer, Font_16x26, YELLOW, BLACK);

sprintf(buffer, "Vout: %lu.%03lu", vout / 1000, vout % 1000);
ST7789_WriteString(10, 240 - 3 * 26, buffer, Font_16x26, YELLOW, BLACK);

sprintf(buffer, "Iout: %lu.%03lu", iout / 1000, iout % 1000);
ST7789_WriteString(10, 240 - 2 * 26, buffer, Font_16x26, YELLOW, BLACK);

sprintf(buffer, "PWM: %3lu",
        (100 * __HAL_TIM_GET_COMPARE(&htim1, TIM_CHANNEL_1)) / 1000);
ST7789_WriteString(10, 240 - 26, buffer, Font_16x26, YELLOW, BLACK);
task_delay(200);
}
```

## PID регулятор

Реализация PID регулятора в точности совпадает с материалом из лекции. Главная сложность была в выборе периода регулирования и коэффициентов.

```
int my_PID(int target, int feedback, int min, int max)
{
    int dU = (int)target - (int)feedback;
    static int prev_dU = 0;
```

```
static float int_part = 0;

static float kP = 0.15f;
static float kI = 0.015f;
static float kD = 0.005f;

int_part += dU * kI;

if (int_part < min)
{
    int_part = min;
}
if (int_part > max)
{
    int_part = max;
}

int out = dU * kP + int_part + kD * (dU - prev_dU);
prev_dU = dU;
if (out < min)
{
    out = min;
}
if (out > max)
{
    out = max;
}
return out;
}
```

Исходные коды проекта приложены к дополнительным материалам отчета.

## 9. Заключение

В ходе выполнения работы получилось реализовать регулируемый источник напряжения с использованием PID регулятора, а подключение мотора позволило продемонстрировать работу устройства при динамическом изменении нагрузки.

## 10. Дополнительные материалы

Демонстрация работы и материалы к отчету расположены в папке на google диск по следующей ссылке: Материалы к Курсовому проекту