

ГУАП

КАФЕДРА № 44

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

доцент, к.т.н., доцент
должность, уч. степень, звание

подпись, дата

О.О. Жаринов
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №9

Разработка микропроцессорного индикатора величины
напряжения

по курсу: СХЕМОТЕХНИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4143

подпись, дата

Д.В. Пономарев
инициалы, фамилия

Санкт-Петербург 2024

1. Цель работы

Изучить методы схемотехнической и программной реализации интерфейсного взаимодействия модулей вычислительных устройств и аналого-цифровых преобразователей. Разработать модель микропроцессорного индикатора величины электрического напряжения и осуществить моделирование его работы.

2. Вариант задания

Вариант 4.

Вар.	1	2	3	4	5	6	7	8	9	10
АЦП	AD1674	ADS7825	LTC1099	LTC1292	LTC1860	MAX1240	MCP3001	MCP3221	TLC548	ADC0801

Рисунок 1- Вариант

3. Описание основных характеристик модели АЦП LTC1292

Аналого-цифровой преобразователь (АЦП) LTC1292 представляет собой 12-разрядный, 8-канальный (или 4-дифференциальный) АЦП с последовательным интерфейсом. Основные характеристики модели LTC1292 включают разрядность, динамический диапазон входных напряжений и время преобразования. Ниже приведено подробное описание этих характеристик.

1. Разрядность

Разрядность: 12 бит

Описание: Разрядность АЦП определяет количество уровней дискретизации, на которые может быть разделен аналоговый входной сигнал. В случае 12-разрядного АЦП LTC1292, входной аналоговый сигнал преобразуется в одно из 4096 (2^{12}) возможных цифровых значений. Чем выше разрядность, тем выше точность преобразования.

2. Динамический диапазон входных напряжений

Диапазон входных напряжений: 0V до 5V (при использовании

питания 5V)

Описание: Динамический диапазон входных напряжений определяет минимальное и максимальное значения аналогового сигнала, которые АЦП может корректно преобразовать в цифровое значение. В случае LTC1292 диапазон входных напряжений составляет от 0 до 5 вольт, что означает, что АЦП может измерять сигналы в этом диапазоне. Если входное напряжение выходит за пределы этого диапазона, результат преобразования будет некорректным.

3. Время преобразования

Время преобразования: 25 микросекунд (typical) на одно преобразование

Описание: Время преобразования – это время, необходимое АЦП для преобразования аналогового сигнала в цифровое значение. Для LTC1292 время преобразования составляет примерно 25 микросекунд на одно преобразование. Это время определяет, как быстро АЦП может обновлять свои данные и, соответственно, с какой частотой можно проводить измерения.

Временная диаграмма обмена микросхемы АЦП типа LTC1292 представлена на рисунке 2.

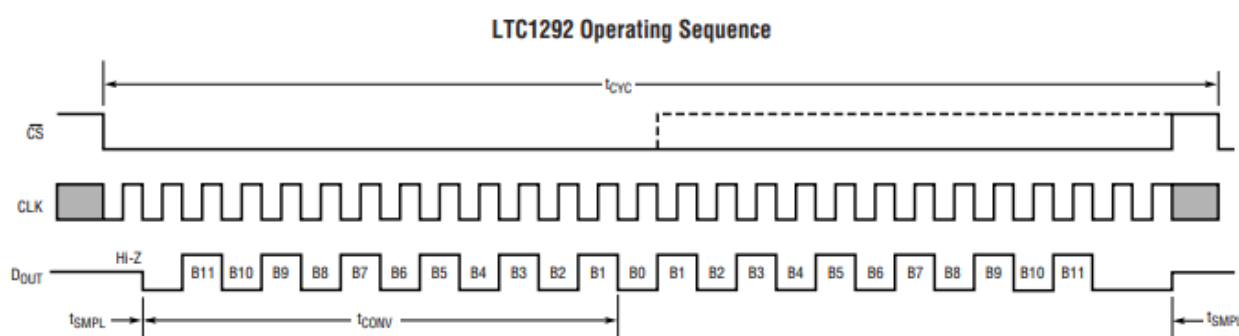
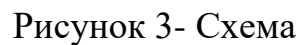


Рисунок 2- Временная диаграмма LTC1292

Далее по аналогии с методическим пособием строим схему с использованием Arduino. Результат представлен на рисунке 3.



Проверка правильности функционирования в этом случае становится самоочевидной – если выходной сигнал повторяет входной (за исключением принципиально неустранимых погрешностей преобразования), значит АЦП функционирует корректно. Результат проверки представлен на рисунке 4.



На рисунке видно, что выходной сигнал повторяет входной.

Далее надо добавить в схему 8 светодиодов и подключить их к свободным портам микропроцессора, сформировав таим образом шкальный индикатор (три старших красного цвета, остальные - зеленые). Результат продемонстрирован на рисунке 5.

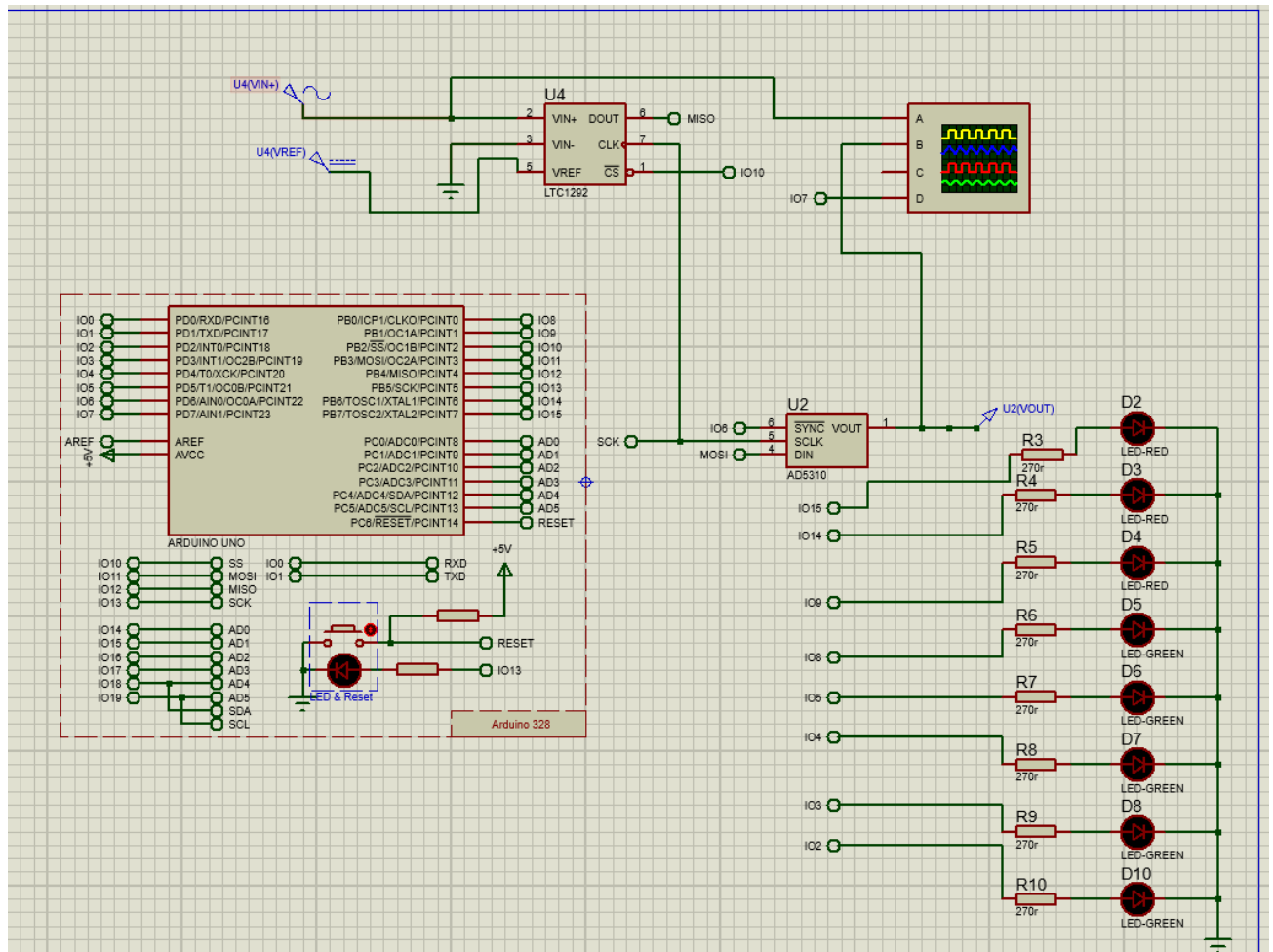


Рисунок 5 – Схема с добавленным шкальным индикатором

Параметры U4(VIN+) и U4(VREF) продемонстрированы на рисунке 6 и 7.

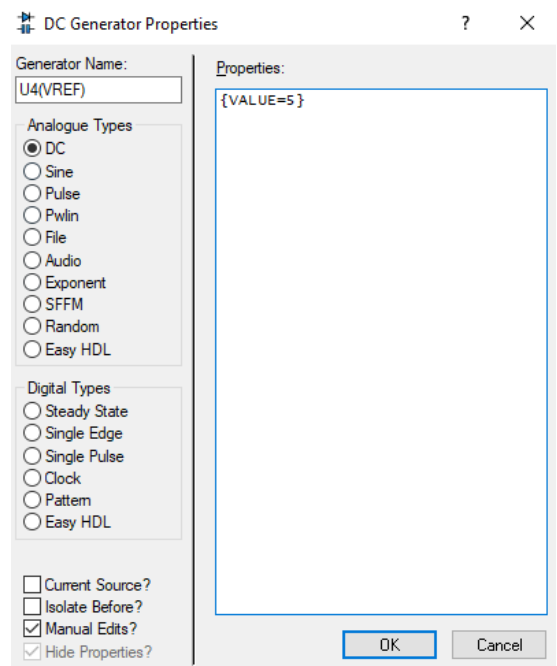
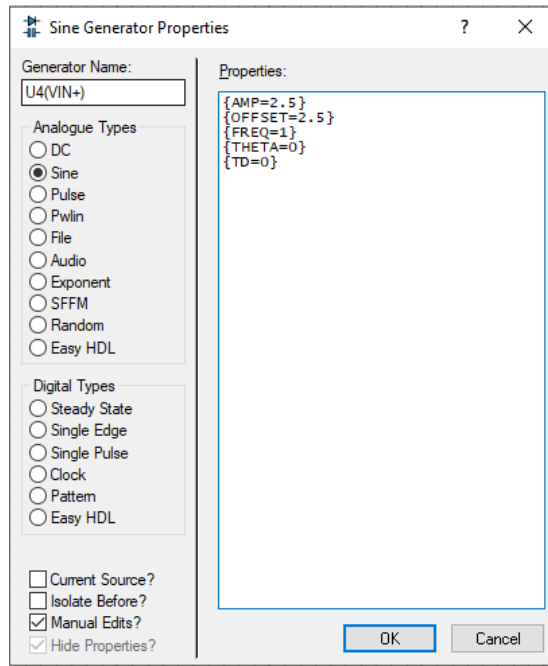


Рисунок 6 – Параметры U4(VIN+) Рисунок 7 – Параметры U4(VREF)

5. Блок-схема программы

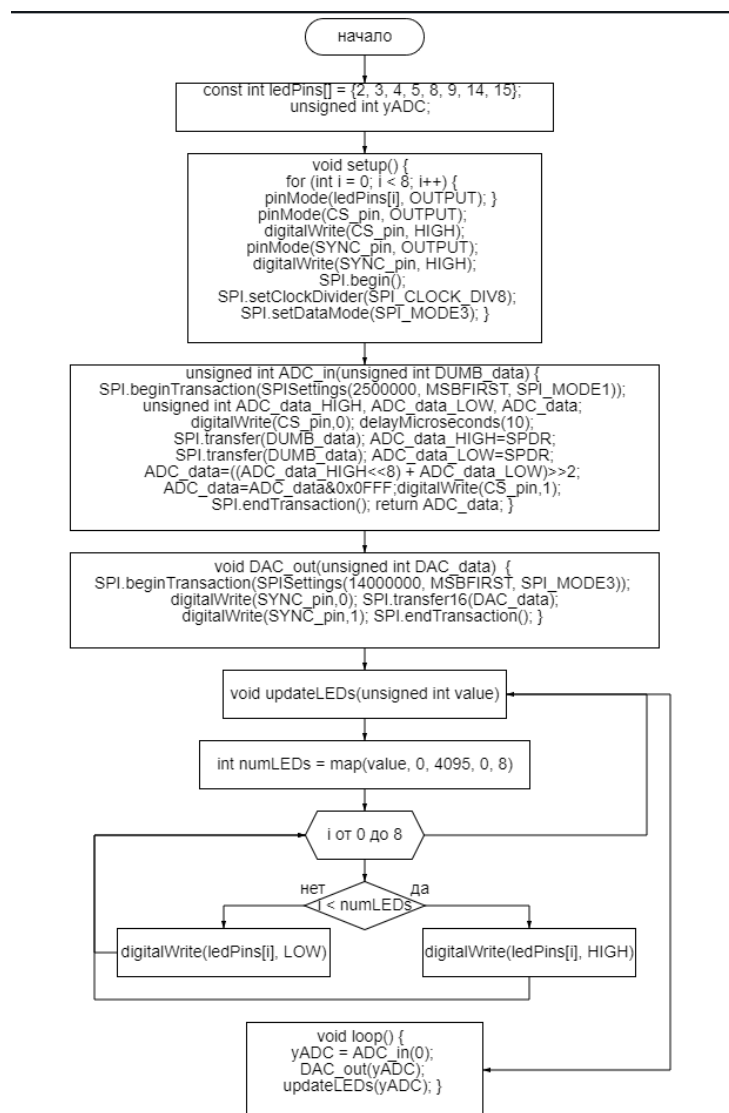


Рисунок 8 –Схема с Signal Generator

6. Текст программы

Программа, написанная для микроконтроллера Arduino, реализует систему измерения и индикации напряжения с помощью аналого-цифрового преобразователя (АЦП) и цифро-аналогового преобразователя (ЦАП). Результаты измерений отображаются на шкальном индикаторе, состоящем из восьми светодиодов.

В функции `setup()` происходит инициализация пинов, к которым подключены светодиоды. Эти пины настраиваются как выходные.

Обновление светодиодов:

Функция `updateLEDs` - эта функция отображает значение, полученное с АЦП, на шкальном индикаторе из светодиодов. Значение с АЦП (от 0 до 4095) преобразуется в количество светодиодов (от 0 до 8) с помощью функции `map()`. Включаются соответствующие светодиоды в зависимости от уровня входного сигнала.

```
/* Main.ino file generated by New Project wizard
 * Created: Чт май 9 2024
 * Processor: Arduino Uno
 * Compiler: Arduino AVR (Proteus)
 */
// Peripheral Configuration Code (do not edit)
//---CONFIG_BEGIN---
#pragma GCC push_options
#pragma GCC optimize ("Os")
#include <core.h> // Required by cpu
#include <cpu.h>
#pragma GCC pop_options
// Peripheral Constructors
CPU &cpu = Cpu;
void peripheral_setup () {
}
void peripheral_loop() {
}
//---CONFIG_END---
#include <math.h> // библиотека математических операций
#include <SPI.h> // библиотека работы с интерфейсом SPI
#define CS_pin 10 // определяем пин для сигнала CS АЦП
#define SYNC_pin 6 // определяем пин для сигнала SYNC ЦАП
// инициализация переменных
const int ledPins[] = {2, 3, 4, 5, 8, 9, 14, 15};
unsigned int yADC; // переменная для кода, получаемого от АЦП
void setup()
{
    // Initialize LED pins as output
    for (int i = 0; i < 8; i++) {
        pinMode(ledPins[i], OUTPUT);
    }
}
```

```

pinMode(CS_pin, OUTPUT); // выход для сигнала CS для АЦП, период следования
импульсов на
//этом выводе равен периоду дискретизации
digitalWrite(CS_pin, HIGH); // чтобы CS АЦП стал 1
pinMode(SYNC_pin, OUTPUT); // выход для сигнала SYNC для ЦАП, период
следования
//импульсов на этом выводе равен периоду дискретизации
digitalWrite(SYNC_pin, HIGH); // чтобы SYNC ЦАП стал 1
SPI.begin(); // инициализация SPI
// настройки будут разными для АЦП и ЦАП поэтому тут ничего
SPI.setClockDivider(SPI_CLOCK_DIV8); // ставим скорость обмена по SPI
SPI.setDataMode(SPI_MODE3); // установка режима SPI
}
// процедура считывания кода из АЦП
// процедура считывания кода из АЦП
// чтобы из этого АЦП что-то получить ему надо отправить
// тактовые импульсы, а они будут образовываться, только если
// передавать какие-нибудь "тупые" (dumb) данные (они появятся на выходе MOSI,
// где их все равно никто воспринимать в это время не будет)
unsigned int ADC_in(unsigned int DUMB_data)
{
SPI.beginTransaction(SPISettings(2500000, MSBFIRST, SPI_MODE1));
unsigned int ADC_data_HIGH, ADC_data_LOW, ADC_data;
digitalWrite(CS_pin, 0); // чтобы CS АЦП стал 0
delayMicroseconds(10);
SPI.transfer(DUMB_data); // отправка тупых данных
ADC_data_HIGH=SPDR;//считывание регистра данных порта SPI
SPI.transfer(DUMB_data); // отправка тупых данных
ADC_data_LOW=SPDR;//считывание регистра данных порта SPI
// сдвиг вправо на 2 разряда нужен, поскольку,
// согласно временной диаграмме обмена для АЦП, последние 2 разряда
неинформативны
ADC_data=((ADC_data_HIGH<<8) + ADC_data_LOW)>>2;
ADC_data=ADC_data&0xFFFF; // выделение 12 бит данных
digitalWrite(CS_pin, 1); // чтобы CS АЦП стал 1 до следующего чтения
SPI.endTransaction();
return ADC_data;
}
void DAC_out(unsigned int DAC_data) // процедура кода в ЦАП
{
SPI.beginTransaction(SPISettings(14000000, MSBFIRST, SPI_MODE3));
digitalWrite(SYNC_pin, 0); // чтобы SYNC ЦАП стал 0
SPI.transfer16(DAC_data); // отправили код - 2 байта - в ЦАП
digitalWrite(SYNC_pin, 1); // чтобы SYNC ЦАП стал 1 до следующей выдачи в ЦАП
SPI.endTransaction();
}
void updateLEDs(unsigned int value) {
    int numLEDs = map(value, 0, 4095, 0, 8); // Map ADC value to 0-8 LEDs
    for (int i = 0; i < 8; i++) {
        if (i < numLEDs) {
            digitalWrite(ledPins[i], HIGH); // Turn on LED
        } else {
            digitalWrite(ledPins[i], LOW); // Turn off LED
        }
    }
}
void loop()
{
// что получили с АЦП то выдаем в ЦАП без изменений
yADC = ADC_in(0); // ноль - это тупые данные
DAC_out(yADC);
updateLEDs(yADC);
}

```

7. Моделирование схемы проекта микропроцессорного индикатора величины напряжения

Далее на рисунках 9-10 будут приведены результаты работы.

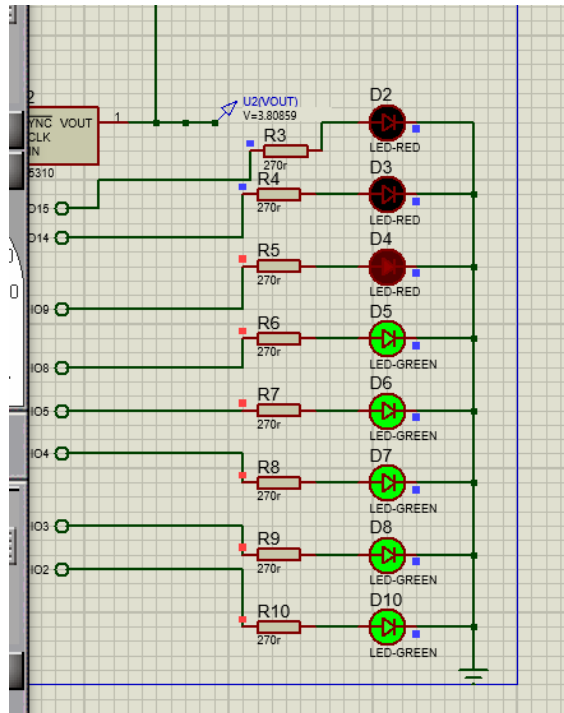


Рисунок 9 – Рез. при 3,80859

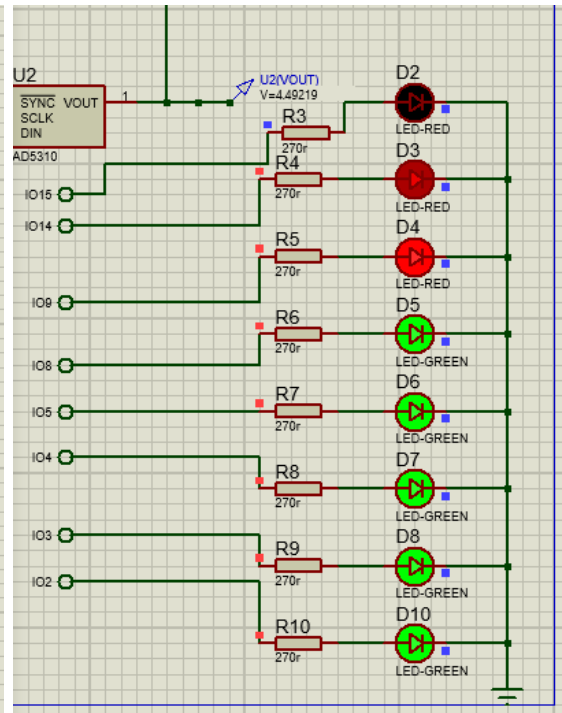


Рисунок 10 –Рез. при 4,49219

Далее я буду подавать на вход АЦП сигнал сложной формы. Для этого я подключу в схему Signal Generator. Результат подключения показан на рисунке 11.

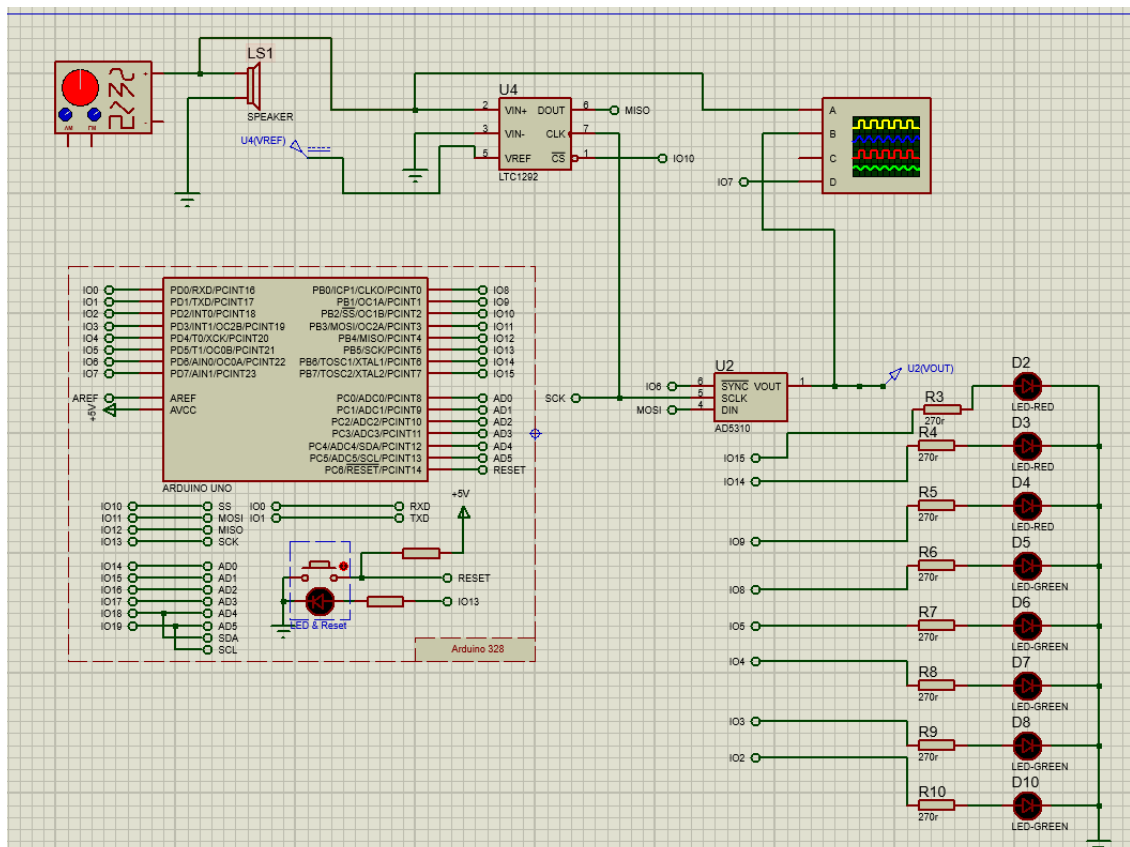


Рисунок 11 –Схема с Signal Generator

Результаты представлены на рисунках 12-15.

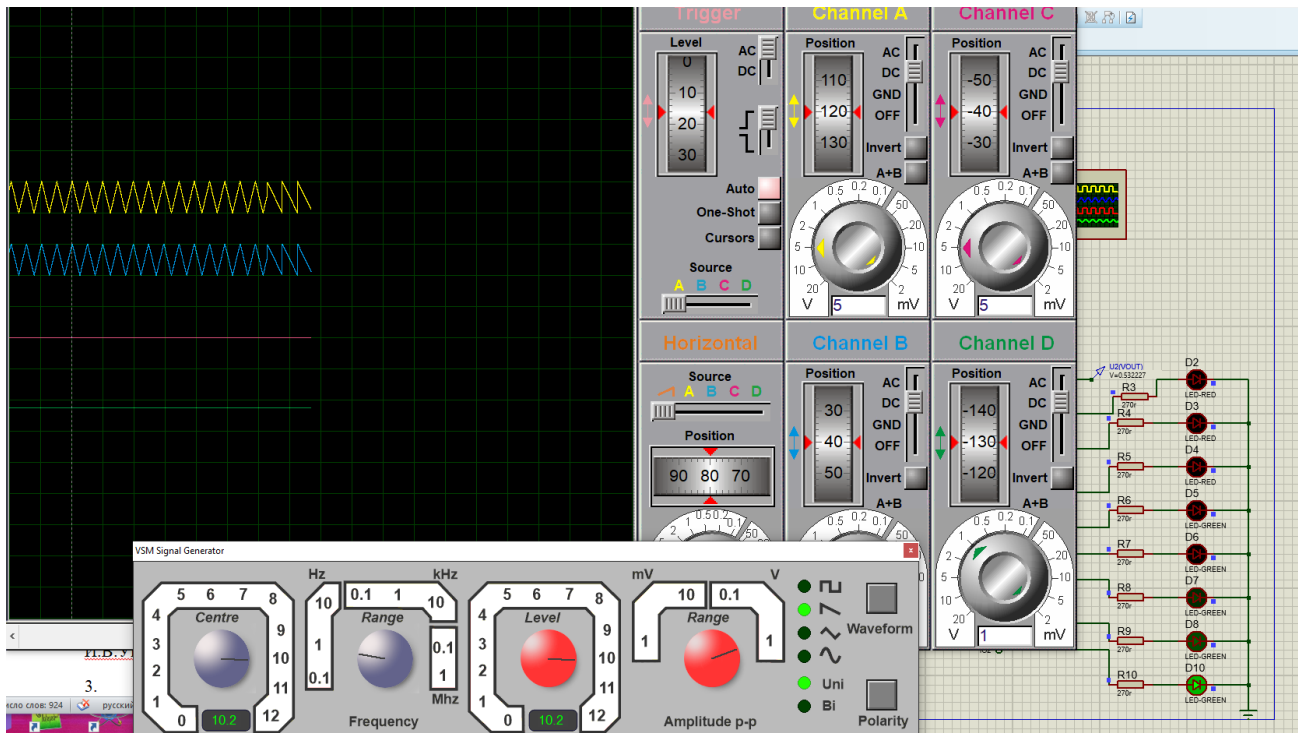


Рисунок 12 –Результат работы(напряжение 0,532277)

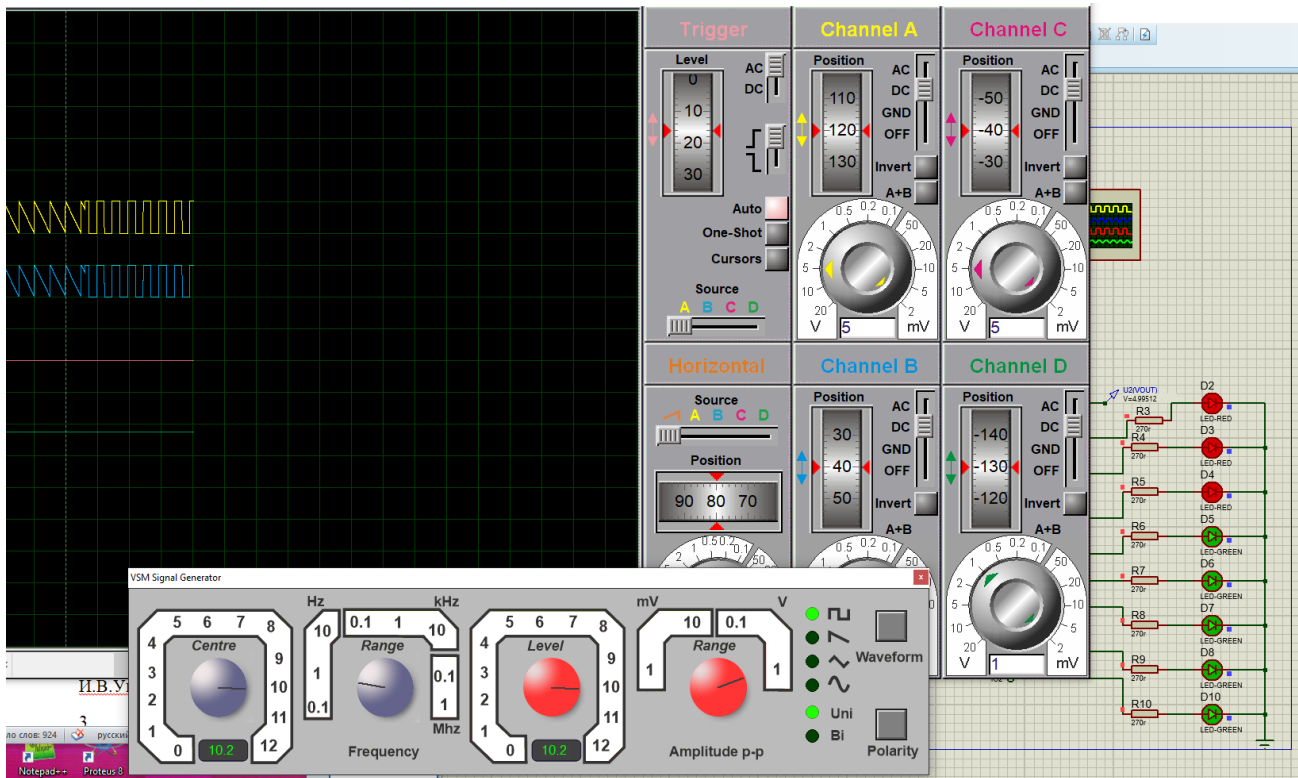


Рисунок 13 –Результат работы (напряжение 4,99512)

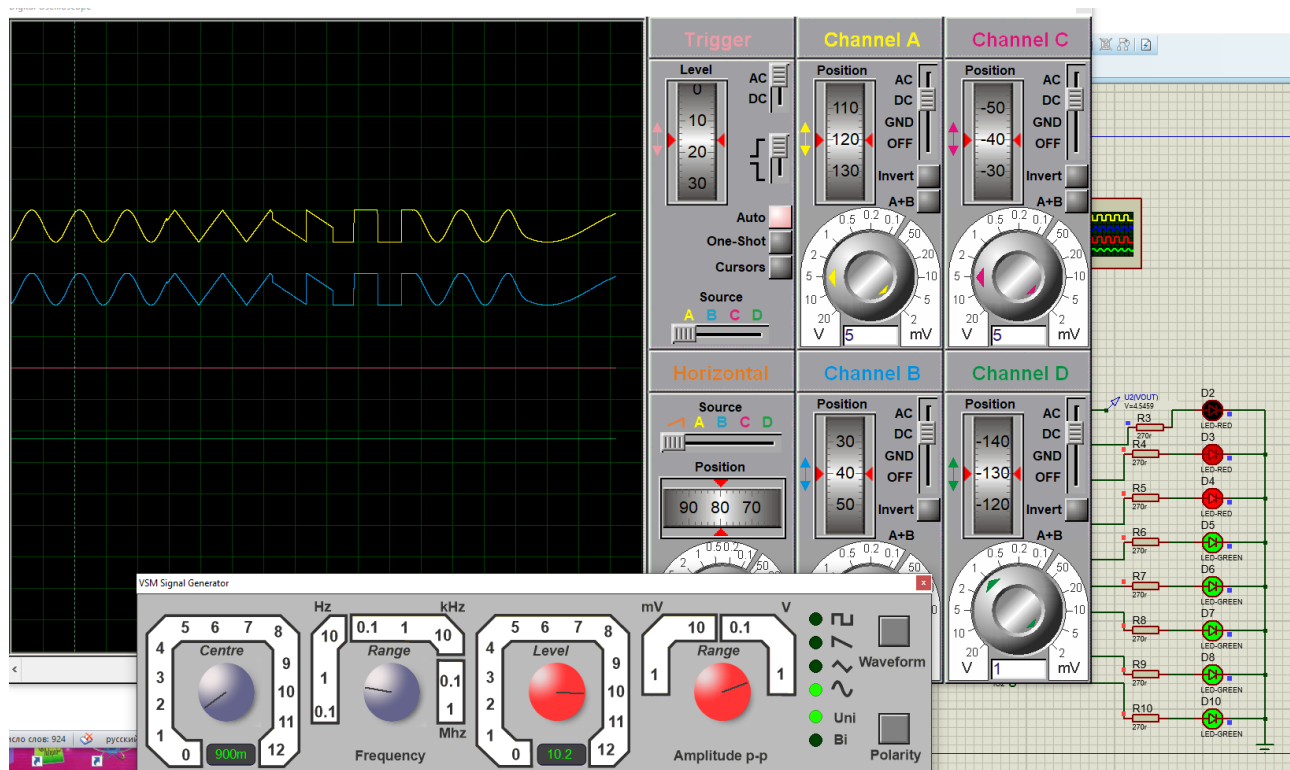


Рисунок 14 –Результат работы(напряжение 4,5459)

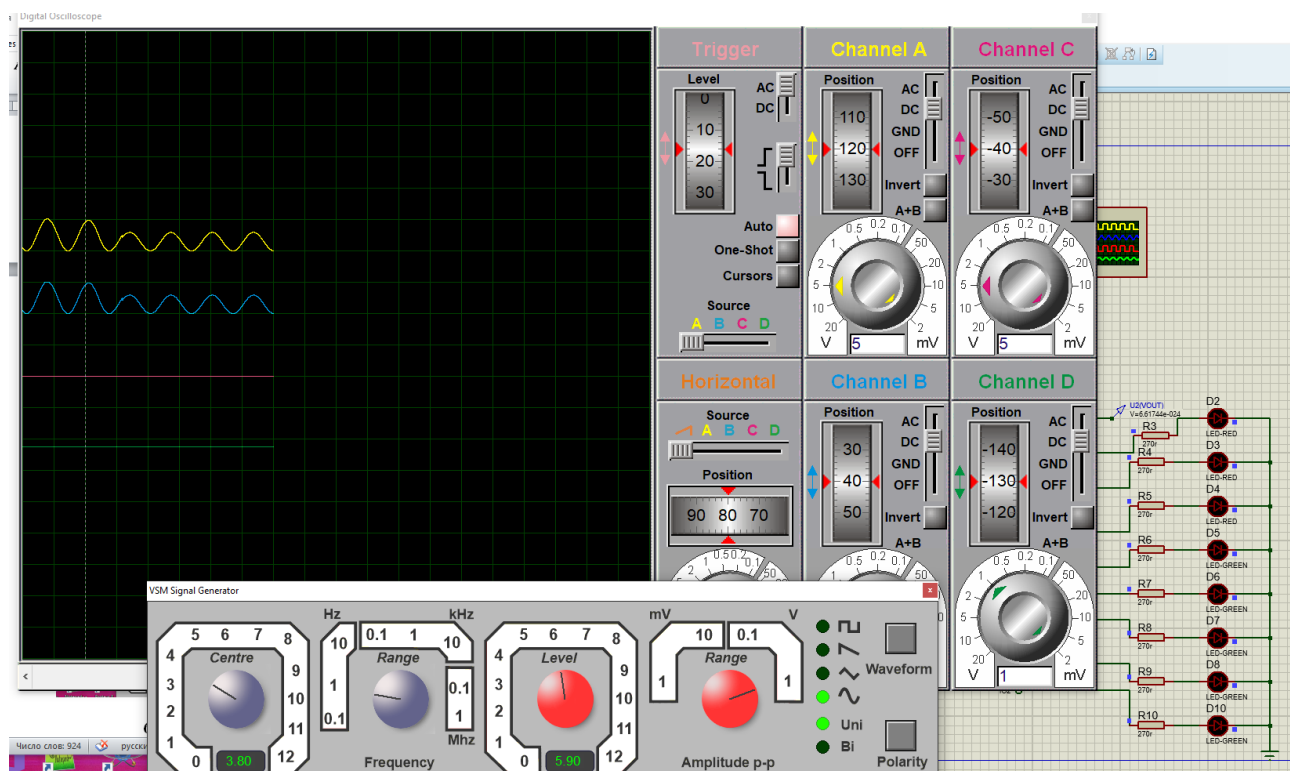


Рисунок 15 –Результат работы (напряжение отр.)

8. Вывод

Изучил методы схемотехнической и программной реализации интерфейсного взаимодействия модулей вычислительных устройств и аналого-цифровых преобразователей. Разработал модель

микропроцессорного индикатора величины электрического напряжения и осуществил моделирование его работы.

9. Список использованных источников

1. Проектирование встраиваемых систем на ПЛИС. / З.Наваби; перев. с англ. В.В. Соловьева. – М.: ДМК Пресс, 2016. – 464 с.
2. Проектирование цифровых устройств на ПЛИС: учеб. пособие / И.В. Ушенина. – СПб: Лань, 2022. – 408 с.
3. Цифровая схемотехника и архитектура компьютера / Д.М. Харрис, С.Л. Харрис; пер. с англ. Imagination Technologies. – М.: ДМК Пресс, 2018. – 792 с.
4. Учебно-методические материалы к выполнению лабораторной работы №9 по дисциплине «Схемотехника» (2-й семестр изучения дисциплины) // Жаринов. О.О: [Электронный ресурс] // Санкт-Петербургский государственный университет аэрокосмического приборостроения. URL.: <https://pro.guap.ru/inside/student/tasks/a3c2a9a2c63d7a16036b6bc107abacdb/download> (Дата обращения: 25.05.24).
5. Документация для Ацп LTC1292. URL.: https://conrad.ru/upload/iblock/bc6/mhhra8py3vyre6naavkt61tpbfi4f1xf/84919e61_0413_11ed_baa7_244bfe93d33a_92cd772b_0413_11ed_baa7_244bfe93d33a.pdf
6. Лекция №6 от 22 апреля 2024 года по дисциплине «Схемотехника» (2-й семестр изучения дисциплины) // Жаринов. О.О: [Электронный ресурс] // Санкт-Петербургский государственный университет аэрокосмического приборостроения. URL.: <https://bbb2.guap.ru/playback/presentation/2.3/98ebb27d1dba4a4bac6ecf053e0ae9024373948a-1713787146024> (Дата обращения: 25.05.24).