

ГУАП

КАФЕДРА № 44

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

доц., канд. техн. наук, доц.
должность, уч. степень, звание

подпись, дата

О.О. Жаринов
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №8

РАЗРАБОТКА И МОДЕЛИРОВАНИЕ МИКРОПОЦЕССОРНОГО
ГЕНЕРАТОРА АНАЛОГОВОГО СИГНАЛА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4143

подпись, дата

Е.Д.Тегай
инициалы, фамилия

Санкт-Петербург 2024

Цель работы

Разработать генератор аналогового сигнала на основе микропроцессора и цифро-аналогового преобразователя и осуществить его моделирование в среде Proteus.

Вариант задания

Содержание индивидуального задания продемонстрировано на рисунках 1 – 2, где на рисунке 1 показан требуемый вариант формы сигнала, а на рисунке 2 – вариант задания на подключение модели ЦАП. Также есть примечание к рисунку 1. N соответствует максимально допустимому значению кода для ЦАП.

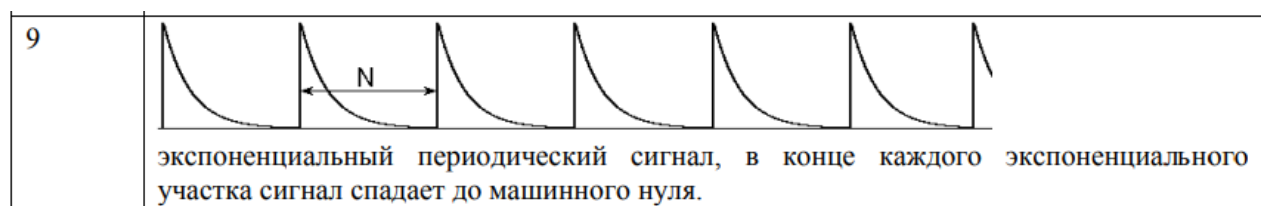


Рисунок 1 – Вариант формы сигнала

LTC1655

Рисунок 2 – Модель ЦАП

Обобщённая структурная схема генератора и описание концепции проектирования

Обобщённая структурная схема генератора представлена на рисунке 3, где цифрой 1 обозначен задающий генератор, 2 – преобразователь (в данном случае – ЦАП), 3 – выходное устройство (в данном случае – осциллограф).

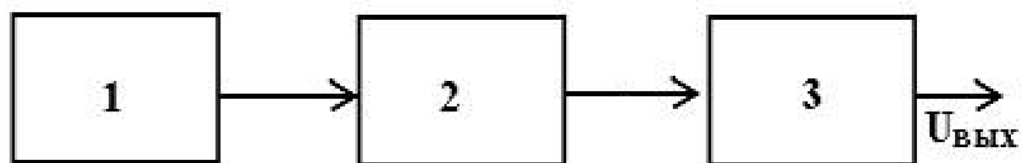


Рисунок 3 – Схема генератора

Задающий генератор отвечает за формирование сигнала с определёнными параметрами. В данной лабораторной работе используется генератор на базе микроконтроллера. ЦАП принимает цифровой сигнал от задающего генератора и преобразует его в аналоговый. Осциллограф

принимает аналоговый сигнал от ЦАП и отображает его на экране в виде временной диаграммы.

Временная диаграмма интерфейса ЦАП

На рисунке 4 изображена временная диаграмма интерфейса ЦАП серии LTC1655.

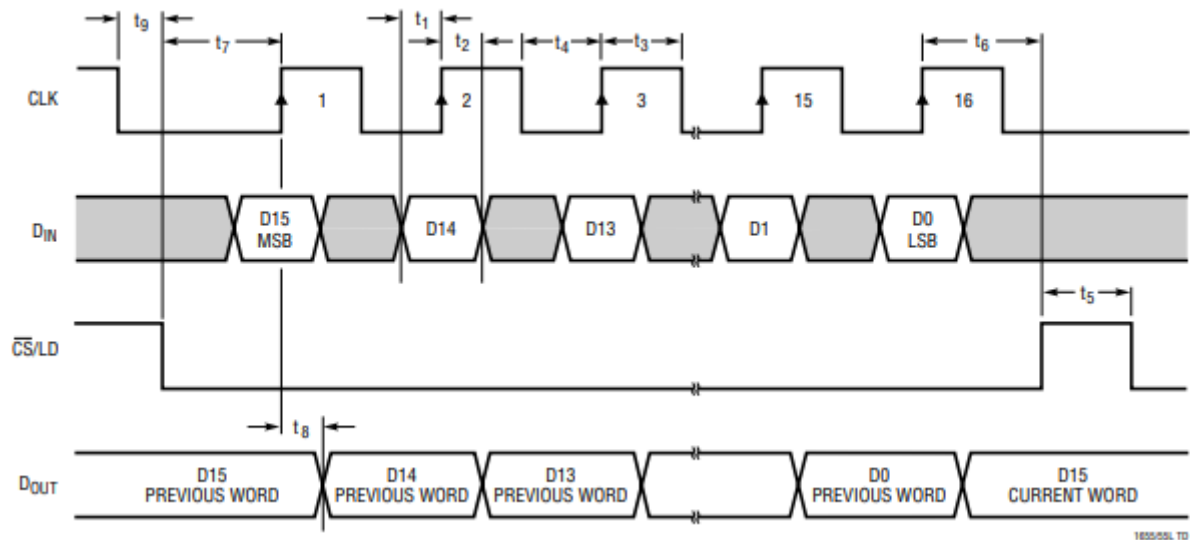


Рисунок 4 – Временная диаграмма

Рассмотрим её подробнее.

- CLK – тактовый сигнал, который управляет временной последовательностью операций внутри ЦАП.
- DIN – этот сигнал представляет собой входные цифровые данные, которые поступают на ЦАП для преобразования в аналоговый сигнал.
- \overline{CS}/LD – сигнал, который используется для указания ЦАП, что сейчас передаются данные на вход
- DOUT – сигнал, обозначающий выходные данные из самого ЦАП.
- D15-D0 – порты ЦАП.
- MSB – указание на то, что порт D15 является старшим битом
- LSB – указание на то, что порт D0 является младшим битом
-

Схема в среде Proteus

При создании проекта была выбрана плата семейства Arduino. Это

показано на рисунке 5.

Обобщённая структурная схема генератора продемонстрирована на рисунке 6.

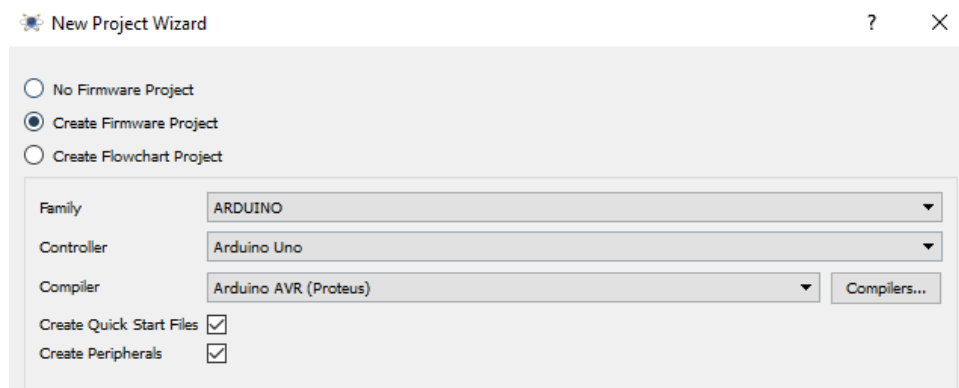


Рисунок 5 – Выбор семейства, контроллера и компилятора

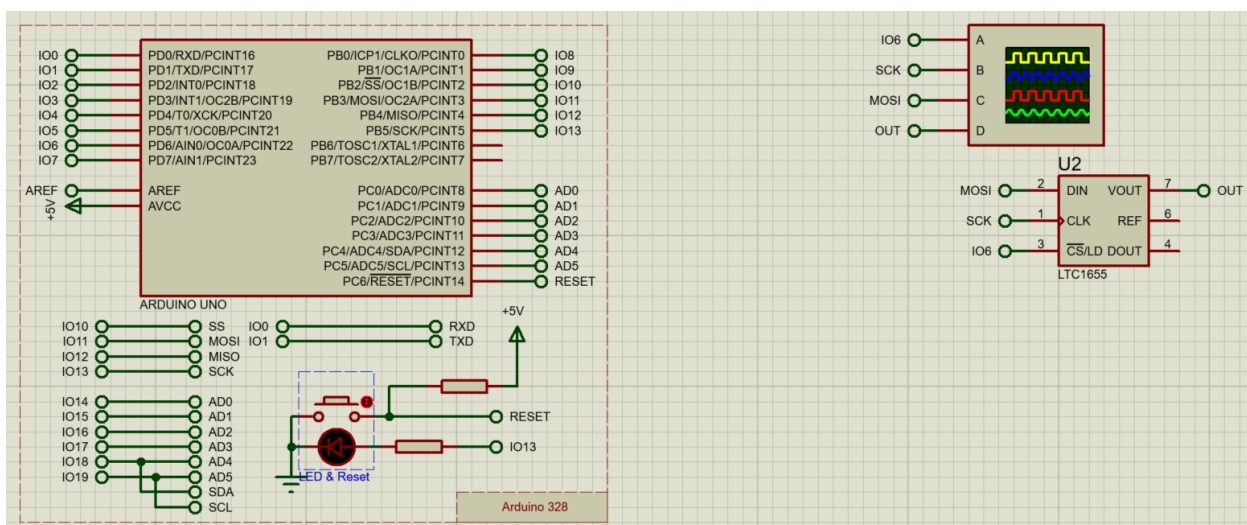


Рисунок 6 – Схема генератора

При создании схемы выбор подключений на каждый из входов и выходов осуществлялся на основе официальной документации используемого ЦАП, которая доступна в Интернете.

Так, например на вход А осциллографа идёт провод Ю6, который представляет собой управляющий сигнал, на вход В - SCK - тактовый сигнал, который генерируется микроконтроллером и синхронизирует передачу. На вход С - MOSI – провод, предназначенный для отправки данных от микроконтроллера к устройству, подключённому по интерфейсу SPI, а на вход D - OUT – выходной сигнал (в контексте данной лабораторной работы это и есть требуемый на выводе сигнал).

Рассмотрим входы и выходы у ЦАП. К порту DIN подключается MOSI. Вообще это порт, который обозначает вход для данных. Он принимает цифровые данные, которые потом ЦАП преобразует в соответствующий аналоговый сигнал. К порту CLK подключается SCK. Этот порт обозначает тактовый сигнал. Он используется для синхронизации передачи данных между микроконтроллером и ЦАПом. К порту \overline{CS}/LD подключается IO6. Этот порт обозначает сигналы Chip Select или Load. Первый сигнал указывает на выбор конкретного устройства на шине данных. Если он активен, то ЦАП готов к приему данных и наоборот. Второй сигнал используется в контексте управления данными. Он может указывать на начало или окончание загрузки новых данных в ЦАП (в разработанной схеме используется первый сигнал). К порту VOUT подключён выходной OUT. Он обозначает выходной аналоговый сигнал ЦАП.

Блок-схема программы

Основные функции, описанные по-отдельности и блок-схема всей программы продемонстрированы на рисунках 7 – 10.

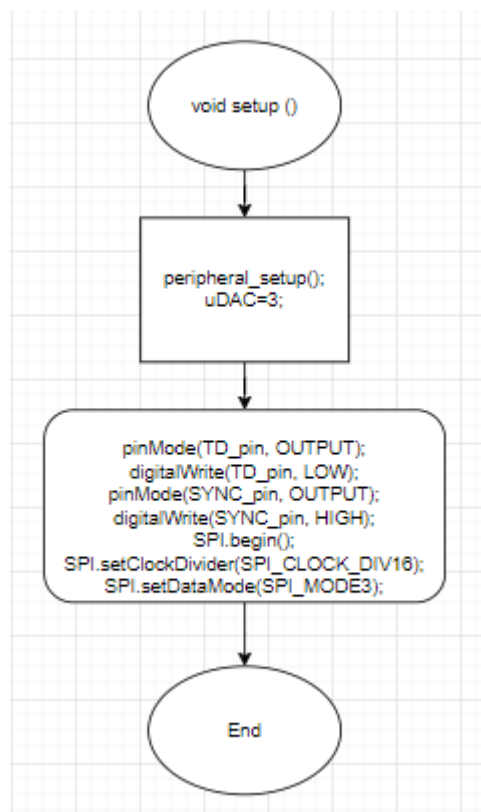


Рисунок 7 – Функция setup

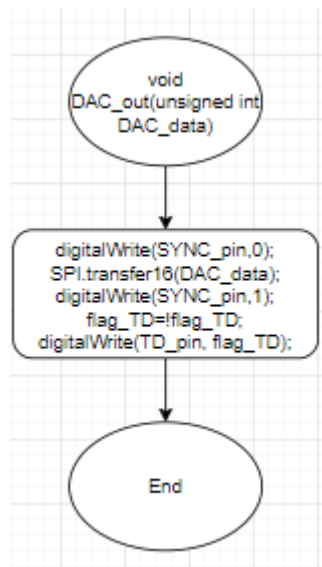


Рисунок 8 – Функция DAC_out

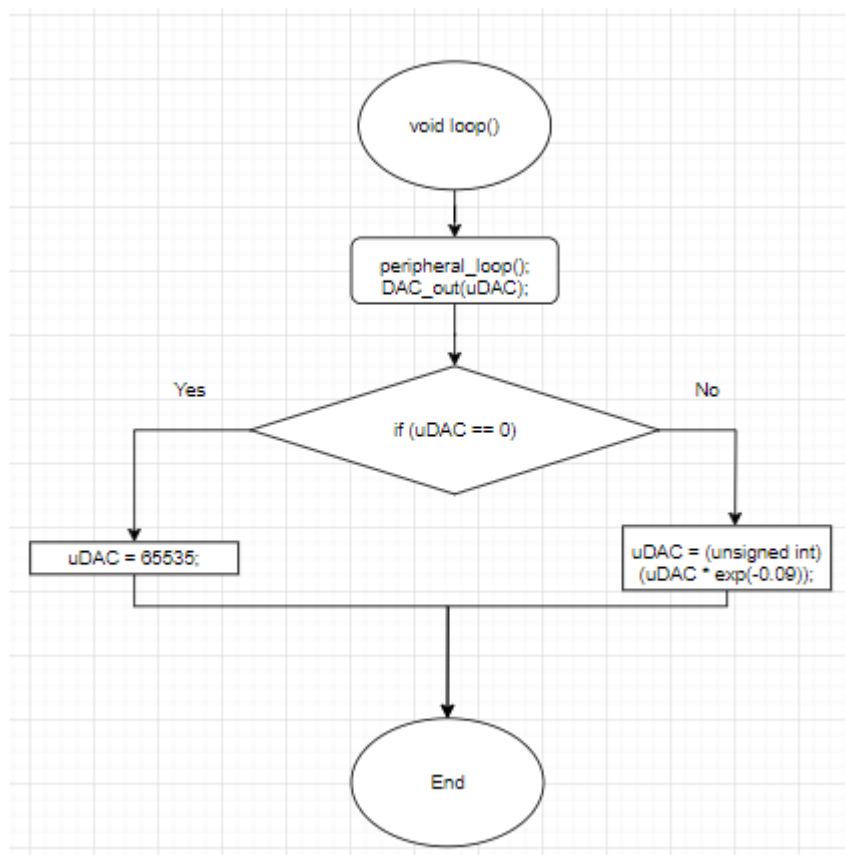


Рисунок 9 – Функция loop

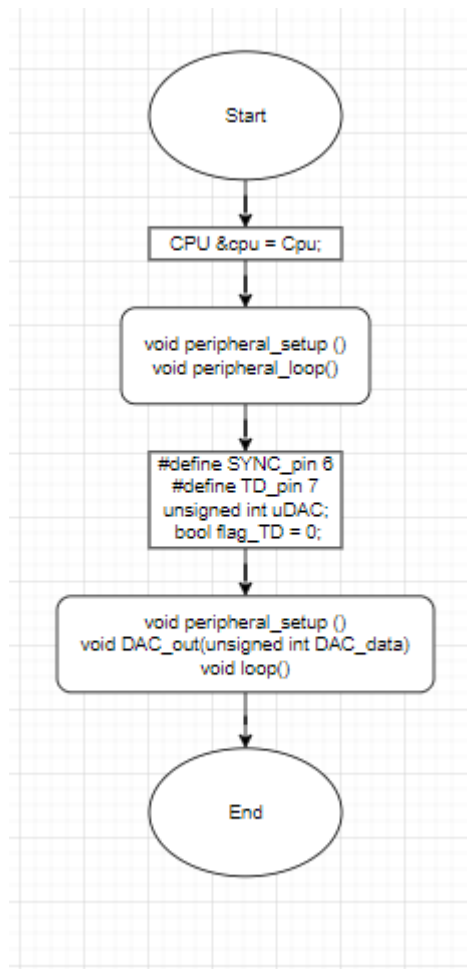


Рисунок 10 – Блок-схема программы

Текст программы

В качестве языка программирования был выбран C.

В начале идёт подключение директив, которые сохраняют текущие опции компилятора GCC во внутреннем стеке компилятора, указывают компилятору GCC оптимизировать код для размера, а также восстанавливают предыдущие опции компилятора GCC. Также идёт подключение файлов-заголовков в проект. Эти файлы содержат основные определения и функции, необходимые для работы с микроконтроллером Arduino Uno, а также определения, специфичные для процессора, такие как регистры и битовые маски для доступа к периферийным устройствам микроконтроллера, таким как порты ввода-вывода, аналоговые компараторы, таймеры и так далее.

После этого идёт подключение уже библиотек, которые содержат математические функции и которые предоставляют функции для работы с интерфейсом последовательной периферии на микроконтроллере Arduino.

Затем идёт определение пина для сигнала SYNC ЦАП и для выхода TD, вспомогательного флага, а также целочисленной переменной uDAC, которая будет отправляться в ЦАП.

После этого идёт обязательная функция setup в рамках использования Arduino. В ней инициализируются различные компоненты и настройки, необходимые для работы программы. Затем идёт функция DAC_out, которая предназначена для записи данных в ЦАП через интерфейс SPI. Наконец, идёт определение функции, которая определяет основной цикл программы. Она и генерирует аналоговый сигнал, который затем отправляется на ЦАП для последующей обработки. Следует отметить, что внутри этой функции используется коэффициент, который был определён отношением последующего относительно предыдущего значения (так как необходимо было добиться длины N между максимальным и минимальным значениями).

Искомый текст программы продемонстрирован ниже.

```
/* Main.ino file generated by New Project wizard
*
* Created: Вс май 12 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 по которому

#define SYNC_pin 6 // определяем пин для сигнала SYNC ЦАП
```



```

#define TD_pin 7 // определяем пин для выхода TD, по которому

// инициализация переменных
unsigned int uDAC; // переменная для целочисленных кодов

bool flag_TD = 0; // Вспомогательный флаг
void setup () {
    peripheral_setup();
    uDAC=3;
    pinMode(TD_pin, OUTPUT); // инициализируем пин TD
    digitalWrite(TD_pin, LOW);
    pinMode(SYNC_pin, OUTPUT); // выход для сигнала SYNC для ЦАП
    digitalWrite(SYNC_pin, HIGH); // чтобы SYNC ЦАП стал 1
    // здесь настройка интерфейса SPI для работы с ЦАП
    SPI.begin(); // инициализируем SPI
    // ставим скорость обмена по SPI:
    SPI.setClockDivider(SPI_CLOCK_DIV16); // можно еще DIV2, DIV4, и т.д.
    // определяем режим SPI: пауза SYNC=1, запись данных по спаду SYNC:
    SPI.setDataMode(SPI_MODE3); // Можно еще MODE0, MODE1, MODE2
}
void DAC_out(unsigned int DAC_data) // процедура записи 16-разрядного
{
    digitalWrite(SYNC_pin,0); // чтобы SYNC ЦАП стал 0
    SPI.transfer16(DAC_data); // запись 2-х байтового числа в ЦАП
    digitalWrite(SYNC_pin,1); // чтобы SYNC ЦАП стал 1 до следующей выдачи в
    ЦАП
    // формируем очередной импульс, длительность которого равна периоду
    //для контроля
    flag_TD=!flag_TD;
    digitalWrite(TD_pin, flag_TD);
}

void loop() {
    peripheral_loop();

    DAC_out(uDAC); // Отправка значения uDAC в ЦАП

    if (uDAC == 0) { // Если uDAC равно 0, начинаем "возврат" к наивысшей точке
        uDAC = 65535; // Новое максимальное значение (наивысшая точка)
    }
    else { // Иначе уменьшаем число по линейному затуханию
        uDAC = (unsigned int)(uDAC * exp(-0.09)); // Экспоненциальный спад
    }
}

```

Временная диаграмма (осциллограмма)

Искомая диаграмма продемонстрирована на рисунке 11.

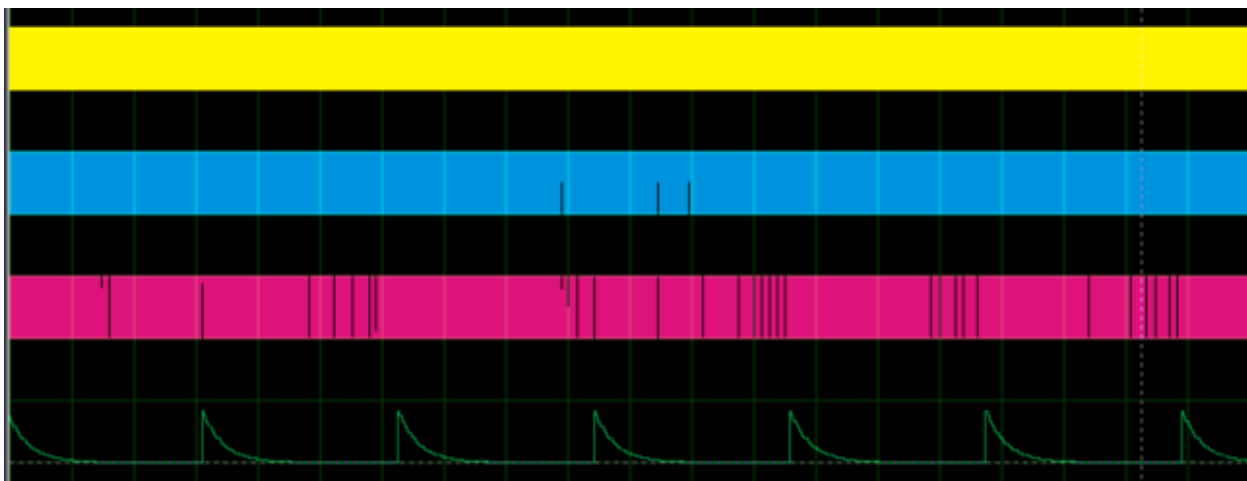


Рисунок 11 - Осциллограмма

Выводы

В данной лабораторной работе был разработан генератор аналогового сигнала на основе микропроцессора и цифро-аналогового преобразователя и также осуществлено его моделирование в среде Proteus.

Список используемых 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. ImaginationTechnologies. – М.: ДМК Пресс, 2018. - 792 с.
4. Методические указания: [Электронный ресурс] //Санкт-Петербургский государственный университет аэрокосмического приборостроения.URL.:
<https://pro.guap.ru/inside/student/tasks/a9e1b6a5105ed976441bdf62cbf78bf8/download>. (Дата обращения: 12.05.24).
5. Лекция №6 от 22 апреля 2024 года: [Электронный ресурс] // Санкт-

Петербургский государственный университет аэрокосмического приборостроения. URL.:

<https://bbb2.guap.ru/playback/presentation/2.3/98ebb27d1dba4a4bac6ecf053e0ae9024373948a-1713787146024>. (Дата обращения: 12.05.24).

6. Лекция №7 от 6 мая 2024 года: [Электронный ресурс] // Санкт-Петербургский государственный университет аэрокосмического приборостроения. URL.:

<https://bbb2.guap.ru/playback/presentation/2.3/d4b490cf4cf7d7ae65c3b14b7d239ccb9e78f2a8-1714996255068>. (Дата обращения: 12.05.24).

7. Официальная документация ЦАП серии LTC1655: [Электронный ресурс]. URL.: <https://www.farnell.com/datasheets/84183.pdf>. (Дата обращения: 12.05.24).

8. Схемотехника телекоммуникационных устройств: учебник / С.И. Зиатдинов, Т.А. Суетина, Н.В. Поваренкин. - М.: Академия, 2013. - 368 с. [Библиотечный шифр 621.396 3-59]

9. X51-совместимые микроконтроллеры фирмы Cygnal: монография / О.И.Николайчук. - М.: ИД СКИМЕН, 2002. - 471 с. [Библиотечный шифр 004 Н63]

10. Основы микропроцессорной техники: учебное пособие / П.Н. Неделин. - СПб: Изд-во ГУАП, 2013. - 63 с. [Библиотечный шифр 004.3(075) Н42]

11. Keil uVision. // URL: <http://cxem.net/software/keil.php>

12. Keil uVision3 V3.51 для платформы MCS51 // URL: <http://microsin.net/programming/MSC51/keil-uvision3-v351.html>

13. Проектирование электронных устройств в Proteus 8.1. Часть 12. // URL: <http://cxem.net/comp/comp204.php>

14. Система моделирования ISIS Proteus. Быстрый старт. // URL: <http://easyelectronics.ru/sistema-modelirovaniya-isis-proteus-bystryj-start.html>

15. Книжная полка: 5 лучших книг о платформе Arduino // URL: <https://arduinoplus.ru/5-knig-ob-arduino/>