

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені І.І.МЕЧНИКОВА

(повне найменування вищого навчального закладу)

Факультет математики, фізики та інформаційних технологій

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних систем та технологій

(повна назва кафедри (предметної, циклової комісії))

Дипломна робота

на здобуття освітньо-кваліфікаційного рівня «бакалавр»

(освітньо-кваліфікаційний рівень)

на тему _____

Розробка цифрового генератора стандартних сигналів

(назва українською)

Development of a digital standart signal generator

(назва англійською)

Виконала: студентка заочної форми навчання

напряму підготовки 6.050102 Комп'ютерна інженерія

(шифр і назва напряму підготовки, спеціальності)

Лепініс Ганна Юхимівна

(прізвище, ім'я, по-батькові)

Керівник ст. викладач Берков Ю.М.

(науковий ступінь, вчене звання, прізвище та ініціали, підпис)

Рецензент д.т.н., професор Гунченко Ю.О.

(науковий ступінь, вчене звання, прізвище та ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри

№ від « » 2021 р.

Завідувач кафедри

Захищено на засіданні ЕК №

протокол № від « » 2021 р.

Оцінка / /
(за національною шкалою, шкалою ECTS, бали)

Голова ЕК

(підпис) Ю.О. Гунченко
(прізвище, ініціали)

(підпис) Н.Ф. Казакова
(прізвище, ініціали)

Одеса – 2021

АНОТАЦІЯ

У даній дипломній роботі розробляється прототип генератору стандартних сигналів з мікроконтролерним керуванням.

Об'єктом дослідження є платформа Arduino та вивчення можливості її використання для цифрового синтезу сигналів стандартних функцій.

Метою роботи є розробка прототипу генератора стандартних сигналів на основі програмувальної плати Arduino. У процесі дослідження спочатку були вивчені методи цифрової генерації сигналів, у тому числі і прямий цифровий синтез (DDS). Після чого була опрацьована ідея й підібрана елементна база для створюваного приладу. Розроблено програмний код, який дозволяє синтезувати сигнали наступних типів: синусоїдальний, прямокутний, трикутний, пилоподібний.

Завершальним етапом є тестування та визначення реальних характеристик генератора, а також виводи про придатність пристрою для використання.

Ключові слова: мікроконтролер AVR, Arduino, цифровий синтез сигналів, DDS, ЦАП, генератор стандартних сигналів, вимірювальні прилади.

АННОТАЦИЯ

В данной дипломной работе разрабатывается прототип генератора стандартных сигналов с микроконтроллерным управлением.

Объектом исследования является платформа Arduino и изучение возможности ее использования для цифрового синтеза сигналов стандартных функций.

Целью работы является разработка прототипа генератора стандартных сигналов на основе программируемой платы Arduino. В процессе исследования вначале были изучены методы цифровой генерации сигналов, включая прямой цифровой синтез (DDS). После чего была проработана идея и подобрана элементная база для создаваемого прибора. Разработан программный код, который позволяет синтезировать сигналы следующих типов: синусоидальный, прямоугольный, треугольный, пилообразный.

Завершающим этапом является тестирование и определение реальных характеристик генератора, а также выводы о пригодности устройства для использования.

Ключевые слова: микроконтроллер AVR, Arduino, цифровой синтез сигналов, DDS, ЦАП, генератор стандартных сигналов, измерительные приборы.

ABSTRACT

In this thesis, the author describes how to develop a prototype of a standard signal generator with microcontroller control.

The subject of research is to study the Arduino board and to explore its capabilities of using for the digital synthesis of standard function signals.

The purpose of this thesis is to develop a prototype of a standard signal generator based on Arduino programmable board. The study first explored the methods of digital signal generations including direct digital synthesis (DDS). Then the concept was designed and the elements of prototype were selected. Software code was developed so that to generate signals of the following types: sine, square, sawtooth or triangular.

The final stage is testing and determination of the real capabilities of generator, and conclusions on the suitability of the prototype usage.

Key words: AVR microcontroller, Arduino, digital signal synthesis, DDS, DAC, standard signal generator, measuring instruments.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	6
ВСТУП	7
1 ФОРМИ СИГНАЛІВ І ТИПИ ГЕНЕРАТОРІВ	8
1.1 Форми коливань	9
1.2 Основні форми сигналів	14
1.3 Типи генераторів сигналів.....	18
2 ЦИФРОВИЙ СИНТЕЗ СИГНАЛІВ.....	20
2.1 Способи генерації сигналів	20
2.2 Метод генерації DDS	23
2.3 Усікання фази	27
3 АНАЛІЗ ТА ПІДБІР КОМПОНЕНТІВ ДЛЯ ПРОЕКТУ	31
3.1 Вибір мікроконтролерної платформи.	31
3.2 Відладочна плата Arduino UNO.....	33
3.3 Модуль енкодера EC11.....	35
3.4 Графічний дисплей Nokia 5110.....	36
3.5 ЦАП на основі резистивної матриці R-2R.....	38
4 СТВОРЕННЯ ТА ТЕСТУВАННЯ ГЕНЕРАТОРА.....	41
4.1 Схема пристрою	41
4.2 Розробка програмного коду	42
4.3 Технічні характеристики генератора	45
4.4 Тестування приладу	47
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
ДОДАТОК А.....	58

ПЕРЕЛІК СКОРОЧЕНЬ

ВЧ – високі частоти

МК – мікроконтролер

НСД – найбільший спільний дільник

ПЗП – постійний запам'ятовуючий пристрій

ЦАП – цифро-аналоговий перетворювач

AVR – родина восьмибітових мікроконтролерів фірми Atmel

DDFS – Direct Digital Frequency Synthesis (прямий цифровий синтез частот)

DDS – Direct Digital Synthesis (прямий цифровий синтез)

EEPROM – Electrically Erasable Programmable Read-Only Memory (постійний запам'ятовувач)

SPI – Serial Peripheral Interface (послідовний периферійний інтерфейс)

SRAM – статична оперативна пам'ять з довільним доступом

ВСТУП

Генератори сигналів відіграють важливу роль при здійсненні електричних вимірювань та випробувань. Вони слугують джерелами тестових сигналів, які подаються на такі випробувані компоненти, як фільтри, підсилювачі або навіть готові модулі з метою перевірити їхню роботу й вивчити їхню поведінка й характеристики.

Історично склалося так, що задачу створення різноманітних сигналів виконують окремі спеціалізовані генератори сигналів – від генераторів надчистих аудіосигналів до багатогигагерцових генераторів радіочастотних сигналів. І хоча існує безліч приладів, що серійно випускаються, найчастіше користувачу для вирішення своїх завдань доводиться доробляти існуючий генератор або самому виготовляти спеціалізований генератор. Проектування вимірювального генератора – дуже складна задача.

На щастя, технологія дискретизації й обробки сигналу дала нам можливість вирішити практично будь-яку задачу за допомогою одного єдиного приладу – генератора сигналів довільної форми.

З іншого боку, розвиток напівпровідникових технологій зробив доступними досить високопродуктивні мікроконтролери, що дозволяють як обробляти, так і синтезувати електричні сигнали в широкому частотному діапазоні.

Об'єктом дослідження є платформа Arduino та вивчення можливості її використання для цифрового синтезу сигналів стандартних функцій.

Метою роботи є розробка прототипу генератора стандартних сигналів з мікроконтролерним керуванням. Генератор повинен генерувати сигналів наступних типів: синусоїдальний, прямокутний, трикутний, пилоподібний. Він має забезпечувати високоточне регулювання вхідної частоти та мале спотворення сигналів.

1 ФОРМИ СИГНАЛІВ І ТИПИ ГЕНЕРАТОРІВ

Генератор сигналів формує електричні сигнали із заданими часовими характеристиками. Залежно від типу генератора сигналів формований сигнал може змінюватися від простого синусоїдального, пилкоподібного й прямокутного до сигналу з аналоговою модуляцією, такими як АМ, ЧМ і ФМ. Діапазон частот може змінюватися від декількох кГц до десятків ГГц. За допомогою використання зовнішнього помножувача частоти можна отримати сигнал частотою до декількох сотень ГГц. Частота вихідного сигналу, як правило, може змінюватися з дуже маленьким кроком (< 1 Гц).

Генератори сигналів можна розділити на два основні класи [1]:

- аналогові генератори сигналів;
- векторні генератори сигналів.

Генератори сигналів, в основному, використовуються при розробці й виробництві електронних модулів і компонентів. Сигнал, формований генератором, подається на випробуваний модуль (підсилювач, фільтр і т.п.). Потім вихідний сигнал модуля аналізується за допомогою відповідного контрольно-вимірювального обладнання, наприклад аналізатора спектра або сигналів, осцилографа, вимірника потужності і т.д. На основі результатів такого аналізу можна визначити, чи коректно модуль виконує свої функції. Крім стандартних функцій вибору частоти, амплітуди й режиму модуляції, сучасні генератори сигналів також мають спроможність додавати шум до тестового сигналу або імітувати багатопроменеве розповсюдження (завмирання) вхідного сигналу. Таким чином, з'являється можливість вивчити роботу приймача при наявності сигналів.

Хоча в точному значенні генератори сигналів не є вимірювальним обладнанням, завдяки описаним вище функціям вони все-таки є тестовими передавачами.

1.1 Форми коливань

Характеристики коливань

«Форму коливань» можна визначити, як послідовність мінливих значень, що повторюється з певною періодичністю [2]. У природі коливання зустрічаються на кожному кроці: звукові коливання, біоструми мозку, океанські хвилі, світлові хвилі, коливання напруги й багато чого іншого. Усі вони являють собою періодично повторювані процеси. Звичайно генератори сигналів створюють електричні коливання (як правило, коливання напруги), які повторюються із заданою періодичністю.

Час повного повторення хвилі називається «періодом». Форма коливань є графічним представленням хвильового процесу – зміни його в часі. Графік хвилі напруги являє собою традиційну діаграму в декартових координатах, по горизонтальній осі якої відкладений час, а по вертикальній – напругу. Деякі прилади можуть захоплювати або створювати хвилі струму, потужності й інших величин.

Амплітуда, частота й фаза

Форми коливань мають багато характеристик, але основними є амплітуда, частота й фаза.

Амплітуда: визначає значення напруги сигналу. Амплітуда перемінного сигналу постійно змінюється (у вітчизняній літературі амплітудою називають розмах або максимальне значення перемінної величини). Генератори сигналів дозволяють задати діапазон напруг, наприклад, від -3 до $+3$ вольтів. У результаті буде генеруватися сигнал, що змінюється між цими двома значеннями, причому швидкість його зміни залежить від форми хвилі й від частоти.

Частота: швидкість повторення періодичного сигналу. Частота вимірюється в Герцах (Гц), або в періодах у секунду. Частота є зворотною величиною від періоду коливань, який дорівнює часові між двома

аналогічними значеннями в сусідніх коливаннях. Чим вище частота, тем менше період.

Фаза: теоретично, це положення початку періоду коливань стосовно точки 0 градусів. Практично, фазою називають зсув початку періоду коливання стосовно опорного коливання або опорної мітки часу (див. рис.1.1).

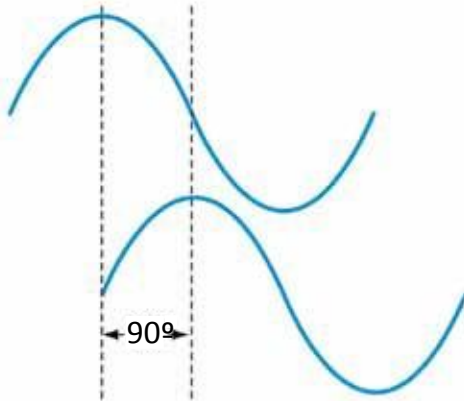


Рисунок 1.1 – Зсув фази (відомий як затримка) описує зсув за часом між двома сигналами. Звичайно фаза виражається в градусах (як показано на малюнку), але в деяких випадках зручніше використовувати час

Найпростіше пояснити поняття фази на прикладі синусоїдального сигналу. Рівень напруги синусоїдального сигналу математично пов'язаний із круговим рухом. Подібно повному оберту, один період синусоїди рівний 360 градусам. Фазовий кут синусоїди описує минулу частину повного періоду.

Два сигнали можуть мати однакову частоту й амплітуду, але відрізнятися по фазі. Зсув фази (відомий також, як затримка) описує зсув за часом між двома аналогічними в усіх інших відносинах сигналами, як показано на рис. 1.1. В електроніці фазові зсуви є широко розповсюдженим показником.

Амплітуда, частота й фаза є основними характеристиками сигналу, використовуваними генератором для оптимізації сигналу відповідно до

вимог практично будь-якого застосування. Крім того, є й інші характеристики, що додатково визначають сигнал, які також можуть регулюватися в багатьох генераторах.

Тривалість фронту й спаду

Тривалості фронту й спаду звичайно описують поведінку імпульсних і прямокутних сигналів. Ці параметри описують час, необхідний сигналу для переходу з одного стану в інший. У сучасних цифрових схемах ці значення вимірюються наносекундами й навіть меншими величинами.

Тривалість фронту й спаду вимірюється як час проходження сигналу між точками, що відповідають 10% і 90% амплітуди (у деяких випадках використовуються точки 20% і 80%). На рис. 1.2 показаний типовий імпульс і деякі його характеристики. Приблизно такий сигнал ви побачите на екрані осцилографа, частота дискретизації якого значно перевищує частоту вхідного сигналу. При більш низьких частотах дискретизації сигнал буде виглядати більш «прямокутним».

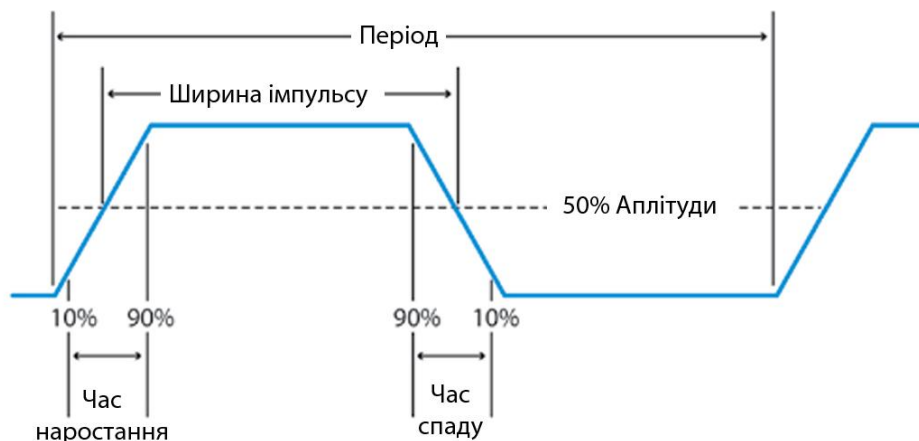


Рис. 1.2 – Основні параметри імпульсу

У деяких випадках тривалості фронту й спаду генеруємих імпульсів

потрібно змінювати незалежно, наприклад, в імпульсах, генеруємих для вимірювання параметрів підсилувача з несиметричною швидкістю відпрацювання, або для керування часом охолодження лазера в установці для точкового зварювання.

Тривалість імпульсу

Тривалістю імпульсу називається інтервал часу між фронтом і спадом. Термін «фронт» може стосуватися як до позитивного перепаду, так і до негативного перепаду, що відноситься й до терміна «спад». Інакше кажучи, ці терміни визначають порядок проходження подій протягом заданого періоду, сама ж полярність імпульсу не впливає на визначення його фронту або спаду. Тривалість імпульсу визначається часом між точками, що відповідають 50% амплітуди фронту й спаду.

Інший термін, «коефіцієнт заповнення», використовується для опису відносної тривалості високого й низького рівня імпульсу. Наприклад, на мал. 5 показаний імпульс із коефіцієнтом заповнення 50% (у вітчизняній літературі використовується поняття «шпаруватість» – відношення періоду імпульсу до його тривалості, тобто імпульс на рис. 1.2 має шпаруватість 2.). На відміну від цього, прямокутний сигнал з періодом 100 нс, активний високий рівень якого триває 60 нс, буде мати коефіцієнт заповнення 60% .

Для кращого розуміння коефіцієнта заповнення уявіть собі виконавчий механізм, який для запобігання перегріву двигуна подає на двигун односекундний імпульс, після чого робить паузу на три секунди. У результаті такий привід простоює три секунди з кожних чотирьох – тобто коефіцієнт заповнення в цьому випадку рівний 25%.

Зсув

Не всі сигнали змінюються симетрично щодо землі (0 В). Напругою зсуву називають напругу між нульовим рівнем і центральною лінією сигналу. По суті, напруга зсуву дорівнює постійній складовій сигналу, що складається

із суми перемінної й постійної напруги, як показано на рис. 1.3.

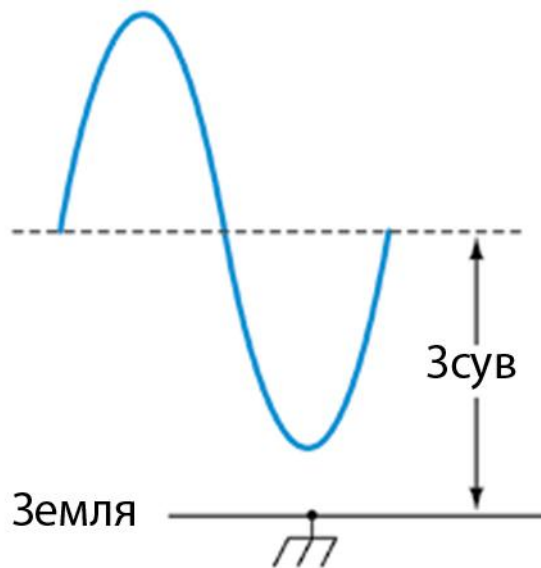


Рисунок 1.3 – Зсув описує постійну складову сигналу, що складається із суми постійної й перемінної напруги

Диференційні й несиметричні сигнали

Диференційними сигналами називаються сигнали, що використовують два комплементарні сигнальні тракти, що передають копії того самого сигналу прямої і зворотної полярності (щодо землі). Якщо протягом періоду сигналу одного тракту стає більш позитивним, то сигнал іншого тракту стає рівною мірою більш негативним. Наприклад, якщо в деякий момент часу значення сигналу в одному із трактів рівно $+1,5$ В, то значення сигналу в іншому тракті буде рівно $-1,5$ В (за умови ідеальної синфазності сигналів). Диференційна лінія добре придушує зовнішні наведення й шуми, пропускаючи тільки корисні сигнали.

На практиці більше поширена несиметрична конфігурація, у якій використовується лише один тракт проходження сигналу й земля. На рис. 1.4 показані диференційний і несиметричний способи передачі сигналу.

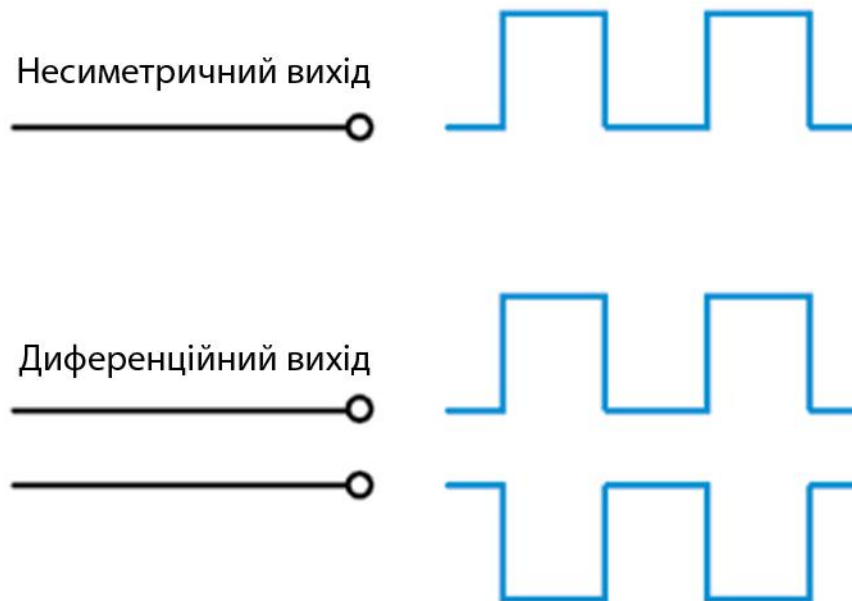


Рисунок 1.4 – Несиметричні й диференційні сигнали

1.2 Основні форми сигналів

Сигнали можуть мати всілякі форми. У більшості електронних приладів використовуються сигнали однієї або декількох описаних нижче форм, найчастіше з додаванням шуму або спотворень:

- синусоїдальні сигнали;
- меандри й прямокутні сигнали;
- пилкоподібні й трикутні сигнали;
- перепади й імпульсні сигнали;
- складні сигнали.

Синусоїдальні сигнали

Синусоїдальні сигнали, імовірно, самі пізнавані із усіх сигналів. Більшість джерел живлення змінного струму виробляють саме синусоїдальні сигнали. У звичайних побутових електричних розетках у будинках присутня напруга синусоїдальної форми. І практично завжди синусоїдальні сигнали

використовуються для демонстрації законів електротехніки в навчальних лабораторіях. Синусоїдальний сигнал описується простою математичною функцією – його форму ідеально визначає крива « $\sin x$ » у межах 360 градусів.

Спеціальним випадком синусоїди є загасаюча синусоїда, яка являє собою загасаючі коливання, що виникають у ланцюгу після подачі на неї імпульсу.

На рис. 1.5 наведені приклади синусоїдального й загасаючого синусоїдального сигналу.

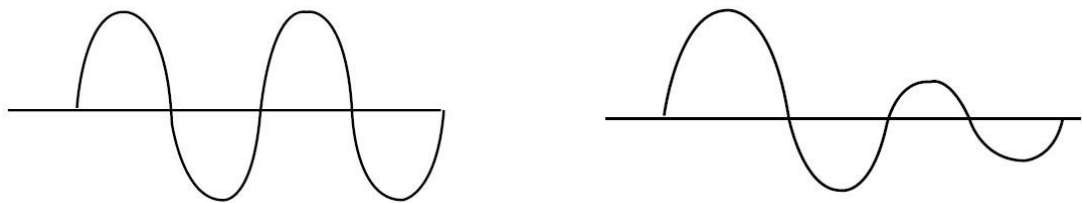


Рисунок 1.5 – Синусоїда й загасаюча синусоїда

Меандри й прямокутні сигнали

Меандри й прямокутні сигнали є базовими сигналами, що складають основу всієї цифрової електроніки, хоча, звичайно, вони знаходять застосування й в інших галузях. Сигнал меандр являє собою напругу, що перемикається між двома фіксованими рівнями через рівні інтервали часу. Звичайно такі сигнали використовуються для перевірки підсилювачів, які повинні обробляти швидкі переходи між двома рівнями напруги (тобто описані раніше фронти й спади імпульсів). Меандр є ідеальним сигналом тактової частоти для цифрових систем – комп'ютерів, бездротових комунікаційних пристроїв, систем ТВ високої чіткості й багатьох інших систем.

Сигнал прямокутної форми аналогічний по характеристиках меандру, за винятком того, що інтервали високого й низького рівня не рівні між

собою, як описано раніше в поясненні терміна «коефіцієнт заповнення». Приклади меандру й сигналу прямокутної форми наведені на рис. 1.6.



Рисунок 1.6 – Меандр і прямокутний сигнал

Пилкоподібні й трикутні сигнали

Пилкоподібні й трикутні сигнали дуже схожі на ті геометричні форми, від яких відбулися їхні назви. Пилкоподібний сигнал у кожному періоді лінійно наростає до пікового значення й потім миттєво спадає. Трикутний сигнал має порівнянні часи наростання й спаду. Часто такі сигнали використовуються для керування іншими напругами, наприклад, в аналогових осцилографах і телевизорах. Приклади пилкоподібного й трикутного сигналу наведені на рис. 1.7.

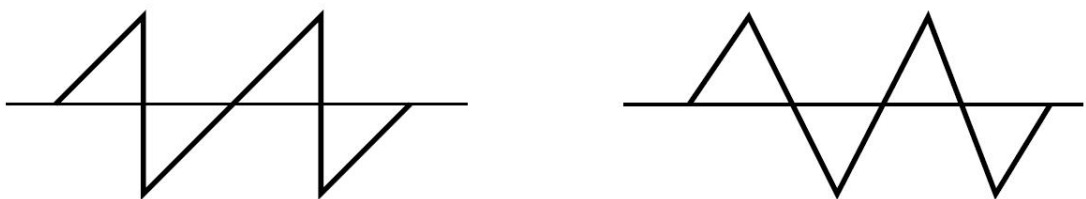


Рисунок 1.7 – Пилкоподібний і трикутний сигнал

Перепади й імпульси

«Перепадом» називають сигнал, що демонструє раптову зміну рівня, наприклад, при замиканні вимикача живлення.

«Імпульс» безпосередньо пов'язаний із прямокутним сигналом. Подібно сигналу прямокутної форми, він утворюється шляхом переходу напруги нагору й потім униз, або вниз і потім нагору, між двома фіксованими рівнями. По своїй природі імпульси являються двійковими сигналами й тому є основним засобом передачі інформації (даних) у цифрових системах. Імпульс може представляти один біт інформації, що проходить через комп'ютер. Група спільно переданих імпульсів утворює пачку імпульсів. Синхронізована група пачок (яка може передаватися паралельно або послідовно) утворює цифрову послідовність. Приклади перепаду, імпульсу й пачки імпульсів наведені на рис. 1.8.

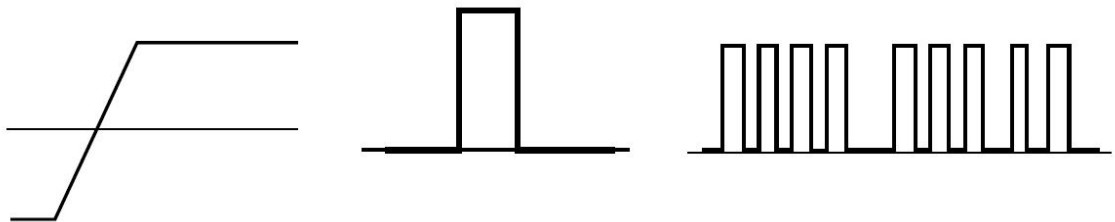


Рисунок 1.8 – Перепад, імпульс і пачка імпульсів

Хоча цифрові дані представлені звичайно імпульсами, сигналами прямокутної форми або меандрами, реальні цифрові сигнали мають більш округлі кути й пологі фронти.

Іноді дефекти ланцюга приводять до спонтанного виникнення імпульсів. Звичайно такі перехідні сигнали носять неперіодичний характер і називаються «глітчями». Одна із проблем налагодження цифрових схем є відділення глітчів від корисних, але вузьких імпульсів даних. І одним з достоїнств деяких типів генераторів є можливість додавання глітчів у пачку імпульсів.

1.3 Типи генераторів сигналів

Вибір відповідного генератора сигналів завжди визначається галуззю застосування. Важливими критеріями є діапазон частот, діапазон рівнів, чистота спектра, доступні типи модуляції (аналогова, цифрова) і можливість додавання певних перешкод до сигналу (шум, імітація багатопроменевого розповсюдження).

Прості генератори низькочастотних сигналів називаються тональними генераторами [3]. Як правило, тональний генератор формує синусоїдальний, прямокутний або пилкоподібний електричний сигнал у чутному людським вухом діапазоні, тобто від 20 Гц до 20 000 Гц. Можна вручну настроїти як частоту, так і амплітуду вихідного сигналу. Тональні генератори найчастіше використовуються в контрольно-вимірювальній апаратурі для акустичних і електроакустичних приладів у комбінації з відповідним вимірником рівня звуку. Наприклад, таким способом можна визначити наявність необхідного рівня на виході передавача тонального сигналу. При вимірюваннях зі змінною вхідною частотою можна додатково використати відповідний вимірник рівня (наприклад, осцилограф, аналізатор спектра, вимірник потужності) для вимірювання повної частотної характеристики передавальної системи, постійно змінюючи вхідну частоту сигналу генератора або використовуючи в якості джерела сигналу аналізатор спектра.

Генератори сигналів, які можуть формувати тільки прості періодичні сигнали, називаються генераторами стандартних сигналів. Як правило, вони можуть формувати синусоїдальні, прямокутні й трикутні коливання. Сучасні генератори стандартних сигналів здебільшого цифрові. Їхня робота заснована на прямому цифровому синтезі (DDS) і вони можуть формувати різні періодичні сигнали. Прикладом використання генераторів стандартних сигналів може слугувати перевірка таких електронних схем, як підсилювачі й фільтри.

Генератори ВЧ-сигналів використовуються для формування

високочастотних синусоїдальних коливань. Також вони часто містять функцію хитання частоти, що дозволяє здійснювати повторюване розгорнення в заданому діапазоні частот. Діапазон частот може досягати від декількох кГц до десятків ГГц. ВЧ-генератори поділяються на дві категорії: аналогові та векторні генератори сигналів. Аналогові генератори сигналів дозволяють формувати сигнали із частотною й амплітудною модуляцією. Вони також можуть формувати імпульсні сигнали. Генератори векторних сигналів до того ж здатні формувати сигнали із цифровою модуляцією для різних стандартів мобільного зв'язку, цифрового радіо й ТВ і т.п. Як і аналогові генератори сигналів, векторні генератори сигналів також можуть модулювати високочастотні несучі сигнали.

Генератори сигналів довільної форми – це векторні генератори сигналів, для яких дані модуляції розраховуються заздалегідь (а не в реальному масштабі часу) і зберігаються в пам'яті приладу. Вихідний сигнал такого генератора завжди є моделюючим. Перевагою таких генераторів є те, що вони способи формувати сигнал практично будь-якої форми й повторювати його практично з будь-якою необхідною частотою. Такі генератори використовуються як універсальні джерела сигналів при розробці нових пристроїв, наукових дослідженнях, випробуваннях і технічному обслуговуванні.

2 ЦИФРОВИЙ СИНТЕЗ СИГНАЛІВ

2.1 Способи генерації сигналів

Існують аналогові й цифрові способи генерації.

Для аналогових способів генерації сигналів використовують автоколивальні системи або автогенератор, принцип якого полягає в тому, щоб енергію від джерела живлення через керуючий елемент подавати в коливальну систему.

Цифрова генерація частот

Основні методи цифрової генерації сигналів – метод апроксимації й табличний метод.

Метод апроксимації використовує представлення періодичного сигналу в аналітичному вигляді й виконує в мікропроцесорі обчислення по заданій програмі. У пам'яті пристрою зберігаються лише параметри сигналу, що генерується. Програма обчислює відліки функції за деяким заданим інтервалом. При необхідності одержання аналогового сигналу далі використовується ЦАП. Або цифрові дані використовуються усередині пристрою, наприклад, в іншій програмі, або для модуляції. Метод апроксимації дозволяє обходитися невеликим об'ємом пам'яті. Однак недоліком є необхідність обчислення значного числа арифметичних і логічних операцій, що затрачуються на формування одного відліку сигналу, що обмежує максимальну частоту сигналу.

Табличний метод генерації передбачає, що попередньо обчислені цифрові відліки функції постійно зберігаються в пам'яті. Формування чергового відліку зводиться до операцій підготовки адреси й читання відповідної комірки пам'яті. Достоїнством цього методу є менший час, що витрачається на формування відліку й, як наслідок, можливість генерації сигналів з більш високою частотою. Недоліком є необхідність мати великий

об'єм пам'яті даних.

До табличного методу відноситься метод прямого цифрового синтезу. Прямий цифровий синтез (від англ. DDS – Direct Digital Synthesizer) – метод, що дозволяє отримати аналоговий сигнал (звичайно це синусоїдальний сигнал, пілкоподібний, послідовність трикутних імпульсів) за рахунок генерації часової послідовності цифрових відліків і їх подальшого перетворення в аналогову форму за допомогою ЦАП. Тому що сигнал спочатку синтезується в цифровій формі, таке обладнання може забезпечити швидке перемикання частоти, високий розв'язання по сітці частот, роботу в широкому діапазоні частот.

Технологія прямого цифрового синтезу використовується для формування коливань із високою якістю й зміни їх параметрів у широкому спектрі найрізноманітніших галузей, наприклад у медицині, промисловості, вимірювальній техніці, системах зв'язку й оборонної галузі.

Ключовою вимогою в багатьох галузях промисловості є спроможність точно формувати коливання різної частоти й форми, легко й швидко змінюючи їх параметри. Незалежно від того, чи вимагається джерело частоти зі швидкою перебудовою, малим фазовим шумом і чудовим вільним від побічних складових динамічним діапазоном для широкопasmового приймача-передавача, або збудливий сигнал зі стабільною частотою для промислової системи керування, можливість швидко, просто й без застосування дорогих засобів генерувати сигнал з регульованими параметрами, підтримуючи при цьому нерозривність його фази, є критичним критерієм проекту, якому задовольняє технологія прямого цифрового синтезу частот.

Задача синтезу частот

Зростаючий попит на більш якісне вимірювальне обладнання, що має менше енергоспоживання, вимагає роботи в нових частотних діапазонах і кращого використання вже наявних. Це породжує необхідність у пошуку більш ефективних способів формування сигналів заданої частоти. У

більшості випадків ця задача вирішується за допомогою синтезаторів частот. Дані пристрою формують із сигналу фіксованої частоти f_{clk} сигнал на пов'язаній з нею бажаній частоті (і з бажаною фазою) f_{out} [4]. У загальному випадку взаємозв'язок може бути описаний простим співвідношенням:

$$f_{\text{out}} = \varepsilon_x \times f_{\text{clk}},$$

де ε_x — масштабуючий множник, який іноді називають нормованою частотою.

Це співвідношення завжди реалізується за допомогою алгоритмів покрокової апроксимації дійсних чисел. Коли масштабуючий множник є раціональним числом, тобто відношенням двох простих цілих чисел, частота вихідного сигналу й опорна частота будуть гармонічно зв'язані один з одним. У той же час у більшості випадків ε_x може належати більш широкому набору дійсних чисел, і тоді процес апроксимації зупиняється, коли результуюче значення множника потрапляє в межі припустимої погрішності.

Цифровий синтез частот

Одним з можливих способів практичної реалізації синтезатора частот є технологія прямого цифрового синтезу частот (Direct Digital Frequency Synthesis, DDFS), яку іноді скорочено йменують прямим цифровим синтезом (Direct Digital Synthesis, DDS). Цей метод заснований на використанні цифрової обробки даних для формування вихідного сигналу з частотою й фазою, що перебудовуються із сигналу фіксованої опорної частоти (тактового сигналу) f_{clk} . В архітектурі DDS опорна частота (частота системного тактового сигналу) піддається діленню на масштабуючий коефіцієнт, який визначається програмувальним двійковим словом настроювання.

Кажучи простою мовою, синтезатор DDS перетворить послідовність імпульсів тактового сигналу в аналогове коливання, як правило,

синусоїдальної, трикутної або прямокутної форми.

2.2 Метод генерації DDS

У найпростішому випадку DDS може бути реалізований як лічильник адрес, на який подаються імпульси з генератора опорної частоти, постійний запам'ятовуючий пристрій (ПЗП) і ЦАП (див. рис. 2.1) [5].

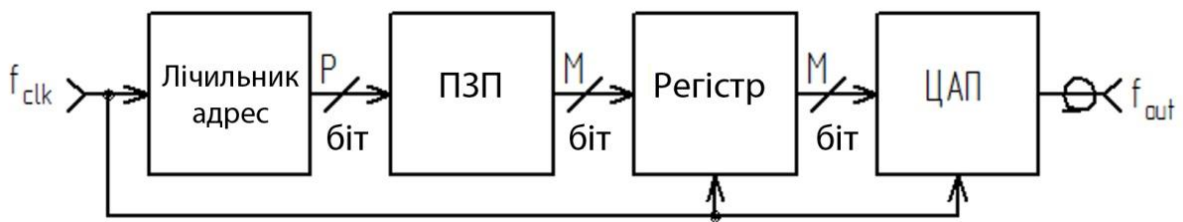


Рисунок 2.1 – Найпростіша функціональна схема DDS

У ПЗП в цифровому вигляді зберігається повний період гармонійного сигналу (синус). ПЗП в цьому випадку функціонує як таблиця синуса. Лічильник адрес послідовно перебирає всі адреси ПЗП, вміст ПЗП через регістр передається на вхід ЦАП. ЦАП у свою чергу для кожного слова із ПЗП генерує сигнал в аналоговій формі. Вихідна частота (f_{out}) такого DDS залежить від двох факторів: по-перше, від опорної частоти f_{clk} , по-друге, від того, з яким кроком розташовані відліки синуса в таблиці ПЗП. Хоча в такої простої архітектури генератора може бути досить висока точність аналогового сигналу й висока продуктивність, але їй не вистачає гнучкості налаштувань. Частота вихідного сигналу може бути змінена тільки при зміні частоти опорного генератора, або перепрограмуванням ПЗП.

Із впровадженням в архітектуру функції фазового акумулятора ядро DDS отримав необхідну гнучкість у керуванні частотою вихідного сигналу. На рисунку 2.2 показана функціональна схема DDS з фазовим акумулятором.

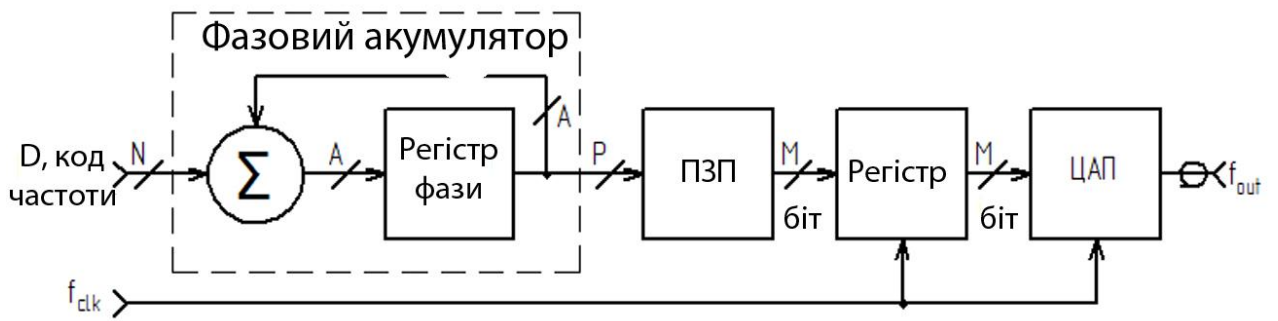


Рисунок 2.2 – Функціональна схема DDS з фазовим акумулятором

Акумулятор фази виконаний за схемою накопичувального суматора і являє собою регістр, який у кожному такті роботи пристрою перезавантажується величиною, рівною старому вмісту, плюс деяка постійна добавка – N розрядний код частоти D . Вміст регістру лінійно збільшується в часі, тільки тепер приріст не завжди є одиничним, а залежить від розміру постійної добавки.

На вхід адрес ПЗП надходять не всі значення регістру фази, а тільки P старших розрядів. Даний механізм називається «усікання фази» і буде розглянутий пізніше.

Функціонування фазового акумулятора може бути показане за допомогою кругової діаграми (рис. 2.3).

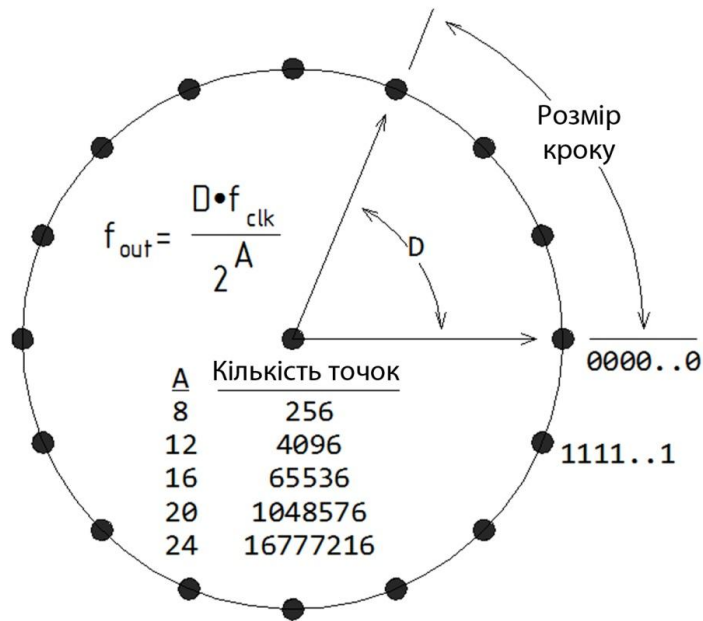


Рисунок 2.3 – Кругова діаграма фазового акумулятора

Кожна точка на круговій діаграмі відповідає еквівалентній точці на циклі синусоїдального сигналу. Число точок на окружності визначається розрядністю фазового акумулятора (A). Фазовий вектор «обертається» з рівномірною швидкістю, кожен цикл обороту вектора відповідає повному періоду гармонійного сигналу. Вихід фазового акумулятора являє собою лінійну періодичну функцію й може бути використаний для безпосередньої генерації пилкоподібного сигналу (якщо подати на вхід ЦАП). Для того щоб отримати гармонійний сигнал вихід акумулятора слід використовувати в якості адреси для ПЗП з таблицею синуса.

Вихідний код акумулятора фази являє собою код миттєвої фази вихідного сигналу. Постійна добавка, яка використовується при роботі акумулятора фази, являє собою приріст фази за один такт роботи пристрою. Чим швидше змінюється фаза в часі, тим більше частота генеруемого сигналу. Тому значення приросту фази фактично є кодом вихідної частоти. Акумулятор фази працює з періодичними переповненнями. Таке періодичне переповнення відповідає періодичній поведінці функції синуса з періодом 2π .

Інакше кажучи, частота переповнень акумулятора фази дорівнює частоті вихідного сигналу. Ця частота визначається формулою:

$$f_{out} = \frac{D \cdot f_{clk}}{2^A}, \quad (1)$$

де f_{out} – вихідна частота;

f_{clk} – тактова частота;

D – код частоти;

A – розрядність акумулятора фази.

При зміні значення D частота на виході синтезатора змінюється відразу, і при цьому сигнал не має розривів. Тут відсутній перехідний процес захоплення частоти, властивий генераторам з петлею ФАПЧ. Важливим параметрами DDS є роздільна здатність по частоті ΔF :

$$\Delta F = \frac{f_{clk}}{2^A}. \quad (2)$$

Із цього співвідношення випливає ще одна унікальна властивість синтезатора частоти: якщо збільшити розрядність N , то зменшиться крок перебудови частоти. Причому особливих обмежень тут немає. Наприклад, якщо розрядність накопичувального суматора 32 біта, а тактова частота становить 50 МГц, той частотне розв'язання складе порядку 0,01 Гц! Для зменшення обсягу ПЗП можна використати властивості симетрії функції синуса. У більшості DDS у ПЗП міститься тільки 1/4 періоду. Правда, при цьому трохи ускладнюється логіка формування адреси.

Основні переваги DDS:

- цифрове керування частотою й фазою вихідного сигналу;
- дуже високе розв'язання по частоті й фазі;
- дуже швидкий перехід на іншу частоту (або фазу), перебудова по частоті без розриву фази, без викидів і інших аномалій, пов'язаних із часом

установлення;

– цифровий інтерфейс легко дозволяє реалізувати мікроконтролерне керування.

2.3 Усікання фази

Усікання фази є важливим аспектом архітектури DDS. Для прикладу розглянемо DDS з 32-розрядним акумулятором фази. Для прямого перетворення 32 біт фази у відповідну амплітуду буде потрібно в ПЗП зберігати 232 відліку синуса. Якщо покласти, що кожен відлік зберігається з 8-розрядною точністю, то буде потрібно 4 гігабайта пам'яті.

Збільшення розрядності акумулятора фази не вимагає обов'язкового збільшення розміру ПЗП. Для адресації можна використати лише необхідну кількість старших розрядів коду фази. Наприклад, в 32-розрядному DDS можна використати тільки старші 12 біт регістру фази. У цьому випадку молодші 20 біт будуть зігноровані (усічені).

Для розуміння наслідків усікання фази повернемося до кругової фазової діаграми. Розглянемо просту архітектуру DDS, що використовує сьома розрядний акумулятор, у якому для адресації ПЗП використовуються старші 5 розрядів. На рис. 2.4 зображена кругова фазова діаграма, сині крапки показують усі можливі значення 7-мі бітного регістру фази (зовнішня окружність), червоні крапки (внутрішня окружність) – значення після усікання – безпосередні адреси для ПЗП. Повний період для акумулятора фази становить 128 кроків, повний період для адрес ПЗП становить 32 кроки.

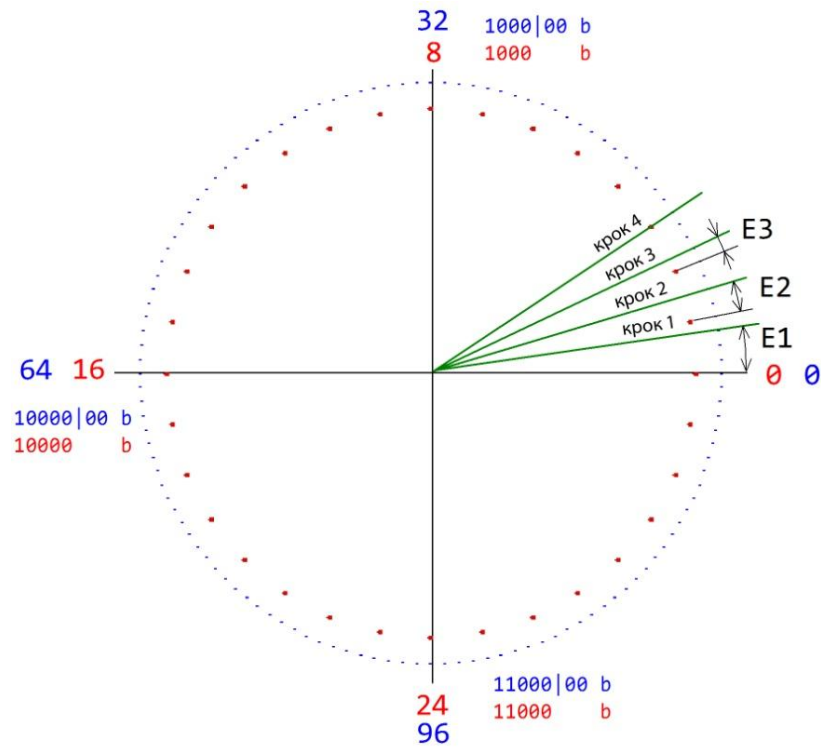


Рисунок 2.4 – Кругова діаграма. Помилки усікання фази

Для 7-ми бітного акумулятора розв'язання фази складе:

$$\frac{360^0}{2^7} = \frac{360^0}{128} \approx 2.81$$

Дозвіл після усікання:

$$\frac{360^0}{2^5} = \frac{360^0}{32} \approx 11.25$$

На круговій діаграмі показані перші чотири фазові кути для коду частоти, рівному 3 (тобто, акумулятор лічить із кроком 3). Зверніть увагу, що перший крок не дотягає до першої мітки внутрішнього кола. У такий спосіб виникає розбіжність між фазою акумулятора (зовнішня окружність) і фазою, обумовленою усіканням до 5-ти біт (внутрішня окружність). Ця розбіжність приводить до помилки в 8.44^0 .

Параметри помилки чотирьох перших кроків акумулятора наведено в таблиці 2.1.

Таблиця 2.1 – Аналіз помилки усікання фази

Номер кроку	Акумулятор		Адреса ПЗП		Помилка
	Значення	Кут	Значення	Кут	
1	3	8.44°	0	0.00°	8.44°
2	6	16.88°	1	11.25°	5.63°
3	9	25.31°	2	22.50°	2.81°
4	12	33.75°	3	33.75°	0.00°

Очевидно, що фазові помилки, внесені усіканням розрядності регістру акумулятора, призведуть до помилок амплітуди в процесі перетворення фази в амплітуду. Незалежно від обраного коду частоти (кроку акумулятора) після достатньої кількості обертів фазового циклу фаза акумулятора й фаза після усікання будуть збігатися. У розглянутому прикладі для цього знадобилося всього чотири кроки. Тому ці помилки є періодичними. А тому що ці помилки періодичні в часовій області, то в спектрі сигналу вони будуть відображатися як окремі лінії (гармоніки).

Рівень паразитних гармонік викликаних ефектом усікання залежить від трьох факторів:

- 1) розрядність акумулятора фази (A);
- 2) розрядність фази після усікання (P);
- 3) значення коду частоти (D).

Очевидно, що при деяких значеннях коду частоти ніякої помилки не відбудеться (для розглянутого прикладу, це $D = 4$) і складові в спектрі не з'являться, у той же час для інших значень D помилка буде максимальною, а рівень паразитних гармонік буде найбільшим.

Якщо різниця A-P більше або рівно 4, що звичайно виконується на практиці, то максимальний рівень паразитної гармоніки (у дБ) щодо основної моди достатньо точно можна знайти по формулі:

$$L_{\text{ph}} = -6.02 * P, \quad (3)$$

де P — розрядність фази після усікання.

Так для DDS з розрядністю акумулятора 32 біта й розрядності фази 12 біт, рівень паразитних гармонік не перевищить -72 дБ.

Значення коду частоти (D), при якому спостерігається найбільший рівень паразитних гармонік, можна визначити по формулі:

$$\text{НСД}(D, 2^{A-P}) = 2^{A-P-1}, \quad (4)$$

де НСД – найбільший спільний дільник.

Для розглянутого прикладу, де $A=7$ і $P=5$, максимальний рівень буде спостерігатися при: $\text{НСД}(D, 2^{7-5}) = 2^{7-5-1}$, $\text{НСД}(D, 4) = 2$.

Тобто, D повинен ділитися без залишку на 2, але не ділитися на 4. Цьому, наприклад, задовольняють значення 2, 6, 10, 14 ...

Значення коду частоти (D), при якому складові відсутні:

$$\text{НСД}(D, 2^{A-P}) = 2^{A-P}. \quad (5)$$

Інакше кажучи, D повинне бути кратне 2^{A-P} . Для розглянутого прикладу D повинне ділитися на 4 без залишку.

3 АНАЛІЗ ТА ПІДБІР КОМПОНЕНТІВ ДЛЯ ПРОЕКТУ

3.1 Вибір мікроконтролерної платформи.

Arduino являє собою невелику плату, з розміщеними на ній компонентами, головним з яких є мікроконтролер сімейства AVR Atmega різних конфігурацій. Він є основною обчислювальною системою цієї платформи, оскільки саме для нього й створюється програмне забезпечення, за допомогою якого мікроконтролер взаємодіє із зовнішнім світом за допомогою спеціальних портів вводу/виводу даних. Даний продукт, складається з економічно вигідних і доступних компонентів і головне його завдання полягає в тому, щоб гарантувати роботу пристрою за принципом «plug-and-play» (підключи й використовуй), — тобто, щоб користувач, діставши плату з коробки й підключивши до комп'ютера, міг негайно приступити до роботи. Управління мікроконтролером користувач здійснює безпосередньо через програмний код. Таким чином, будь-яка плата Arduino насамперед, мікроконтролер, виводи якого зручно розведені по краях плати й підписані. У таблиці 3.1 наведені основні характеристики низки плат Arduino, у таблиці 3.2 – характеристики каналів зв'язку плати з комп'ютером і зовнішніми компонентами. Дані, зазначені в таблицях 3.1 і 3.2 узяті з офіційного сайту розроблювачів сімейства плат і з технічних документацій мікроконтролерів [6].

Поняття, використовувані в таблицях 3.1 і 3.2:

- Electrically Erasable Programmable Read-Only Memory (EEPROM) – пам'ять, що електрично стирається, із можливістю перепрограмування;
- Flash-Пам'ять – пам'ять, яка допускає багаторазовий перезапис усього вмісту;
- Static Random Access Memory (SRAM) – статична енергозалежна пам'ять із довільним доступом;
- Pulse-Width Modulation (PWM) – широтно-імпульсна модуляція

(ШИМ);

– керує середнім значенням напруги на навантаженні через зміни шпаруватості імпульсів;

– Universal Asynchronous Receiver/Transmitter (UART) – універсальний асинхронний приємопередавач.

Таблиця 3.1 – Технічні характеристики програмувальних плат Arduino

Назва	Модель процесора	Вхідне/робоча напруга	Частота процесора	Кількість цифрових виводів(ІО/Р WM)	Кількість аналогових виводів (In/Out)
Mega 2560	Atmega2560	5В/7-12В	16 МГц	54/15	16/0
Uno	Atmega328	5В/7-12В	16 МГц	14/06	6/0
Nano	Atmega328	5В/7-9В	16 МГц	14/06	8/0
	Atmega168				
Due	AT91SAM3X8E	3.3В/7-12В	84 МГц	54/12	12/02
Pro	Atmega328	3.3В/5-12В	8 МГц	14/6	6/0

Таблиця 3.2 – Характеристики зв'язку й пам'яті плат Arduino

Найменування	SRAM (Кбайт)	EEPROM (Кбайт)	Flash (Кбайт)	UART	USB
Mega 2560	8	4	256	4	Regular
Uno	2	1	32	1	Regular
Nano	1	0.512	16	1	Mini-B
	2	1	32		
Due	96	-	32	1	Regular
Pro	2	1	32	1	-

На основі зазначених у таблицях 3.1 і 3.2 даних була обрана плата Arduino UNO.

3.2 Відладочна плата Arduino UNO

Arduino Uno - це пристрій на основі мікроконтролера Atmega328 [7]. У його склад входить все необхідне для зручної роботи з мікроконтролером, його характеристики:

- мікроконтролер Atmega328;
- робоча напруга 5В;
- напруга живлення (рекомендовано) 7-12В;
- напруга живлення (гранична) 6-20В;
- цифрові входи/виходи 14 (з них 6 можуть використовуватися в якості ШІМ-виходів);
- аналогових входів 6;
- максимальний струм одного виводу 40 мА;
- максимальний вихідний струм виводу 3.3V 50 мА;
- Flash-пам'ять 32 КБ (Atmega328) з яких 0.5 КБ використовуються завантажником;
- SRAM 2 КБ (Atmega328);
- EEPROM 1 КБ (Atmega328);
- тактова частота 16 МГц.

Схема розташування елементів і виводів на платі Arduino Uno наведено на рисунку 3.1.

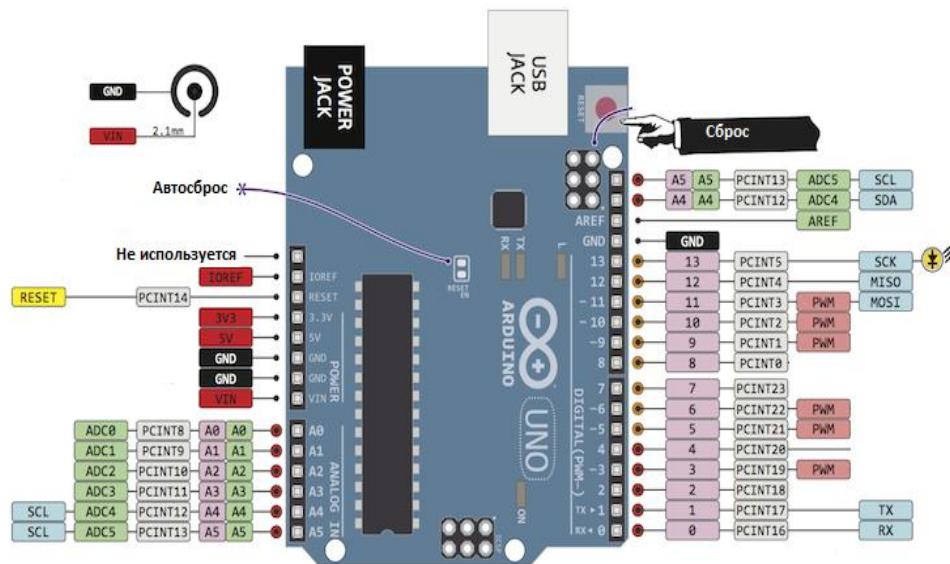


Рисунок 3.1 – Плата Arduino UNO

Входи й виходи. З використанням функцій `pinMode()`, `digitalWrite()` і `digitalRead()` кожний з 14 цифрових виводів може працювати в якості входу або виходу. Рівень напруги на виводах обмежено 5В. Максимальний струм, який може віддавати або споживати один вивід, становить 40 мА. Усі виводи сполучені із внутрішніми резисторами, що підтягують (за замовчуванням відключені) номіналом 20-50 кОм. Крім цього, деякі виводи Arduino можуть виконувати додаткові функції.

Послідовний інтерфейс: виводи 0 (RX) і 1 (TX). Використовуються для одержання (RX) і передачі (TX) даних по послідовному інтерфейсу. Ці виводи з'єднані з відповідними виводами мікросхеми Atmega8U2, що виконує роль перетворювача USB-UART.

Зовнішні переривання: виводи 2 і 3. Можуть служити джерелами переривань, що виникають при фронті, спаді або при низькому рівні сигналу на цих виводах. Для одержання додаткової інформації див. функцію `attachInterrupt()`.

ШИМ: виводи 3, 5, 6, 9, 10 і 11. За допомогою функції `analogWrite()` можуть виводити 8-бітні аналогові значення у вигляді ШИМ-Сигналу.

Інтерфейс SPI: виводи 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Із

застосуванням бібліотеки SPI дані виводи можуть здійснювати зв'язок по інтерфейсу SPI.

Світлодіод: 13. Вбудований світлодіод, приєднаний до виводу 13. При відправленні значення HIGH світлодіод включається, при відправленні LOW - вимикається.

В Arduino Uno є 6 аналогових входів (A0 - A5), кожен з яких може представити аналогову напругу у вигляді 10-бітного числа (1024 різних значення). За замовчуванням, вимір напруги здійснюється щодо діапазону від 0 до 5 В. Проте, верхню границю цього діапазону можна змінити, використовуючи вивід AREF і функцію `analogreference()`. Крім цього, деякі з аналогових входів мають додаткові функції:

TWI: вивід A4 або SDA і вивід A5 або SCL. З використанням бібліотеки Wire дані виводи можуть здійснювати зв'язок по інтерфейсу TWI.

Крім перерахованих на платі існує ще кілька виводів:

AREF. Опорна напруга для аналогових входів. Може задіятися функцією `analogreference()`.

Reset. Формування низького рівня (LOW) на цьому виводі приведе до перезавантаження мікроконтролера. Звичайно цей вивід служить для функціонування кнопки скидання на платах розширення

Мікроконтролер Atmega328 здатний працювати з інтерфейсами I2C (TWI) і Serial Peripheral Interface (SPI). Для зручності роботи із цими інтерфейсами в програмне забезпечення Arduino додана бібліотека Wire, що дозволяє спростити роботу із шиною I2C. Для зв'язку з інтерфейсом SPI використовується бібліотека SPI.

3.3 Модуль енкодера EC11.

Енкодер (від англ. encode – перетворювати) – це пристрій для перетворення кутових положень або лінійних переміщень у цифровий сигнал, тобто Енкодер – це датчик кута або лінійного переміщення, відповідно є

крутильні й лінійні енкодери. Принцип роботи енкодера полягає в перетворенні механічного переміщення в електричні сигнали, у звичайного інкрементального енкодера, який ми будемо розглядати, цей сигнал являє собою два квадратні сигнали (при рівномірному обертанні), зрушених по фазі на 90 градусів.

Модуль енкодера EC11 призначений для побудови систем вводу й керування (рис. 3.2) [8].

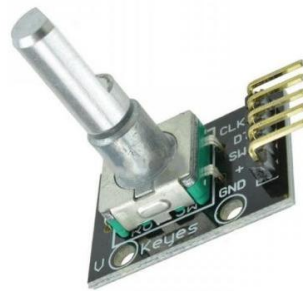


Рисунок 3.2 – Модуль енкодера EC11

Модуль енкодера має 20 фіксованих позицій на один оберт вала. Датчик кута повороту має 3 виводи, позначені буквами А, В і С. Принцип роботи енкодера ґрунтується на зміні положення перемикачів, один з яких відповідає за з'єднання контактів А і В, а другий - за В і С. Кожна зміна положення осі датчика змінює стани перемикачів (рисунок 2.7). Так само, енкодер має кнопку, що спрацьовує при натисканні на вал. Кнопка може мати будь-яке призначення, наприклад, для вибору пунктів меню, зміни чутливості датчика і т.д.

3.4 Графічний дисплей Nokia 5110

Монохромний графічний дисплей 84x48 точки, діагональ 1.6 дюйма з підсвіткою (використовувався в телефонах Nokia 5110) . Побудований на базі

контролера PCD8544 (рис. 3.3) [9].

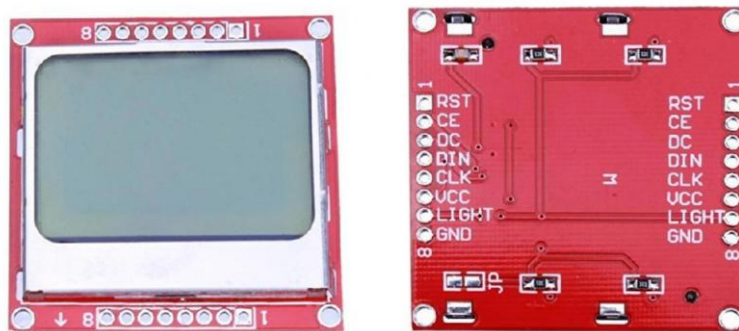


Рисунок 3.3 – Монохромний графічний дисплей

Технічні характеристики дисплея Nokia 5110:

- контролер PCD8544;
- роздільна здатність екрану: 84 x 48 точок;
- розміри екрана: 36 x 24 мм;
- напруга живлення логіки: 2,7 – 3,3В;
- габарити: 43,8 x 43,4 x 13,2 мм;
- вага: 14 г.

Для підключення й передачі даних на дисплеї використовуються два паралельні ряди з 8 контактів. На задній частині дисплея нанесені позначення кожного з них.

Призначення контактів:

- RST — Скидання дисплея, при логічному нулі відбувається скидання зображення на дисплеї;
- CE — вмикання чипа. при логічному нулі чип активний;
- D/C — повідомлення дисплею, чи є дані командою або відображуваними даними;
- DIN — вивід послідовних даних для інтерфейсу SPI;
- CLK — вивід послідовного тактового сигналу для інтерфейсу

SPI;

- VCC — вивід живлення для ЖК-дисплея;
- BL — керує підсвічуванням дисплея;
- GND — земля.

Живлення. На LCD дисплеї 5110 два контакти для живлення. Перший - найважливіший - живлення логіки дисплея. В datasheet зазначено, що воно повинне вибиратися в діапазоні 2.7 - 3.3 В. У нормальному режимі роботи дисплей буде споживати від 6 до 7 мА.

Другий контакт живлення призначений для підсвічування дисплея. Підсвічування реалізоване дуже просто: чотири білі світлодіоди, які розташовані по кутах плати, резисторів, що обмежують струм немає. Можна використати максимальну напругу живлення 3.3 В.

Керуючий інтерфейс. У дисплеї вбудований контролер: Philips PCD8544, який перетворює масивний паралельний інтерфейс у більш зручний серійний. PCD8544 управляється за допомогою синхронної серійного протоколу, що подібний до SPI.

Вище розглянутих серійних контактів установлений ще один – D/C, по якому надходить інформація про те, чи можуть бути відображені дані, які передаються.

3.5 ЦАП на основі резистивної матриці R-2R

Цифро-аналоговий перетворювач (ЦАП) — пристрій для перетворення цифрового (звичайно двійкового) коду в аналоговий сигнал (струм, напруга або заряд). Цифро-аналогові перетворювачі є інтерфейсом між дискретним цифровим світом і аналоговими сигналами [10].

Розглянемо матрицю, зображену на рис. 3.4.

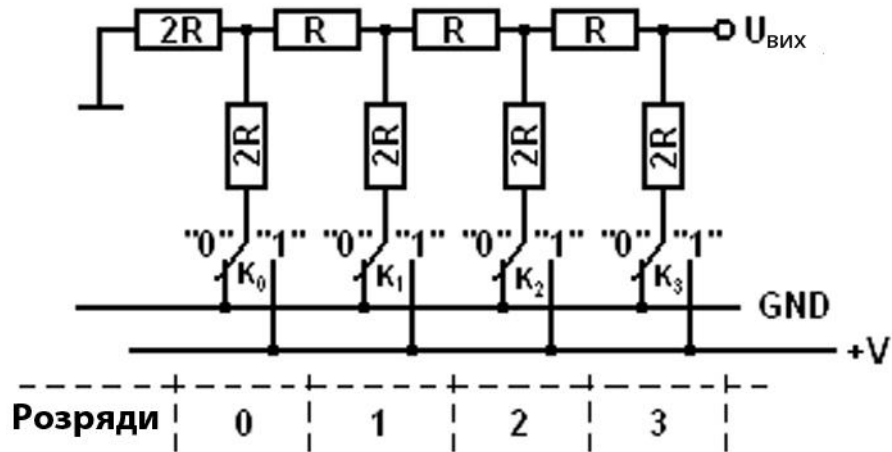


Рисунок 3.4 – 4х розрядна матриця R-2R

Зрозуміло, що якщо всі ключі перебувають у положенні "0", то вихідна напруга буде дорівнювати нулю.

Можна порахувати, що якщо ключ нульового розряду встановлений у положення "1", а всі інші в положення "0", то $U_{вих} = V \cdot 1/16$, якщо ключ першого розряду встановлений у положення "1", а всі інші в положення "0", то $U_{вих} = V \cdot 1/8$, якщо ключі нульового й першого розрядів установлені в положення "1", а всі інші в положення "0", то $U_{вих} = V \cdot (1/16 + 1/8)$, і так далі...

У загальному випадку одержимо: $U_{вих} = V \cdot (A_0 \cdot 1/16 + A_1 \cdot 1/8 + A_2 \cdot 1/4 + A_3 \cdot 1/2)$, де $A_i = 1$, якщо відповідний ключ (K_i) перебуває в положенні "1" і $A_i = 0$, якщо відповідний ключ перебуває в положенні "0".

Тобто, замикаючи різними способами ключі $K_0 \dots K_3$ (або, по іншому кажучи, подаючи на вхід число з 4 біт $A_3 A_2 A_1 A_0$) ми можемо отримати $2^4 = 16$ різних значень вихідної напруги (від $U_{вих} = 0$ до $U_{вих} = V \cdot (1 - 1/16)$ із кроком $\Delta = V \cdot 1/16$).

Таким чином, дана схема являє собою найпростіший паралельний чотирибітний цифро-аналоговий перетворювач.

Аналогічно можна побудувати восьми, десяти, дванадцяти й взагалі n-бітний ЦАП.

У загальному випадку, для n -бітного ЦАП будемо мати: $U_{\text{вих}} = V \cdot \sum (A_i \cdot 1/2^{n-i})$, де i — номер розряду ($i=0, 1, 2 \dots n-1$), $A_i=1$, якщо відповідний ключ замкнути на шину живлення й $A_i=0$, якщо відповідний ключ замкнути на загальний провід. Крок у цьому випадку визначається по формулі: $\Delta = V/2^n$, де n — загальне число розрядів

Зауваження. Коли ми розглядали матрицю $R-2R$, ми вважали ключі ідеальними (опір замкненого ключа дорівнює нулю), але в реальності опір замкнутих ключів (відкритих транзисторів) не дорівнює нулю. Для того, щоб вплив цих паразитних опорів можна було не враховувати, рекомендується вибирати опір R в 2^n (n — розрядність ЦАП) раз більше опору замкнутого ключа [11].

Оскільки в розглянутій схемі немає ніяких інтегруючих ланцюжків, то швидкодія такого ЦАП визначається винятково швидкодією контролера.

Для забезпечення точності перетворень необхідно забезпечити достатню для заданої точності стабільність живлення.

4 СТВОРЕННЯ ТА ТЕСТУВАННЯ ГЕНЕРАТОРА

4.1 Схема пристрою

Як зазначалося раніше, для побудови генератора стандартних сигналів була обрана плата Arduino Uno. Цей вибір обумовлений як задовільними характеристиками мікроконтролера Atmega328 (швидкість роботи, достатня кількість портів, розмір вбудованої пам'яті), так і низькою ціною[12].

Принципова схема пристрою розроблялася за допомогою безкоштовного додатку Fritzing [13] (рис. 4.1).

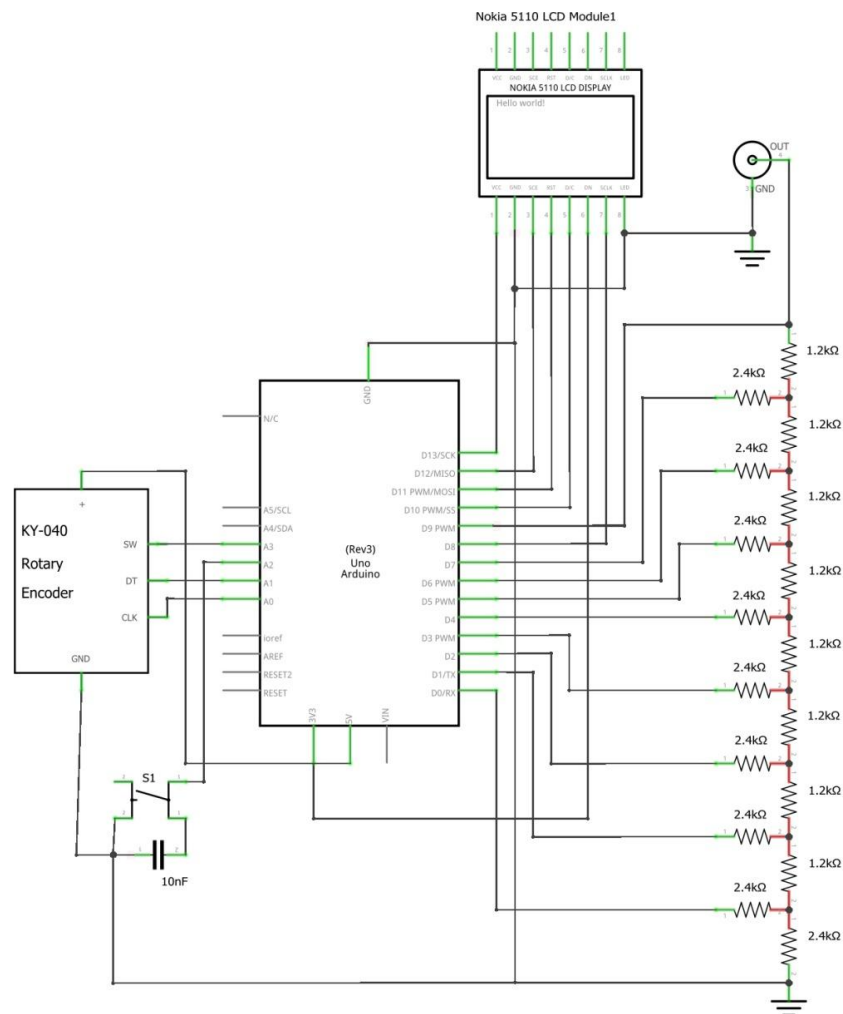


Рисунок 4.1 – Принципова схема генератора стандартних сигналів.

В якості керуючих елементів генератора було застосовано енкодер з кнопкою – регулювання частоти і кроку, та звичайна кнопка для перемикання форми вихідного сигналу. Для відображення налаштувань генератора використано графічний LCD-дисплей Nokia 5110.

Схема зборки генератора наведена на рис. 4.2.

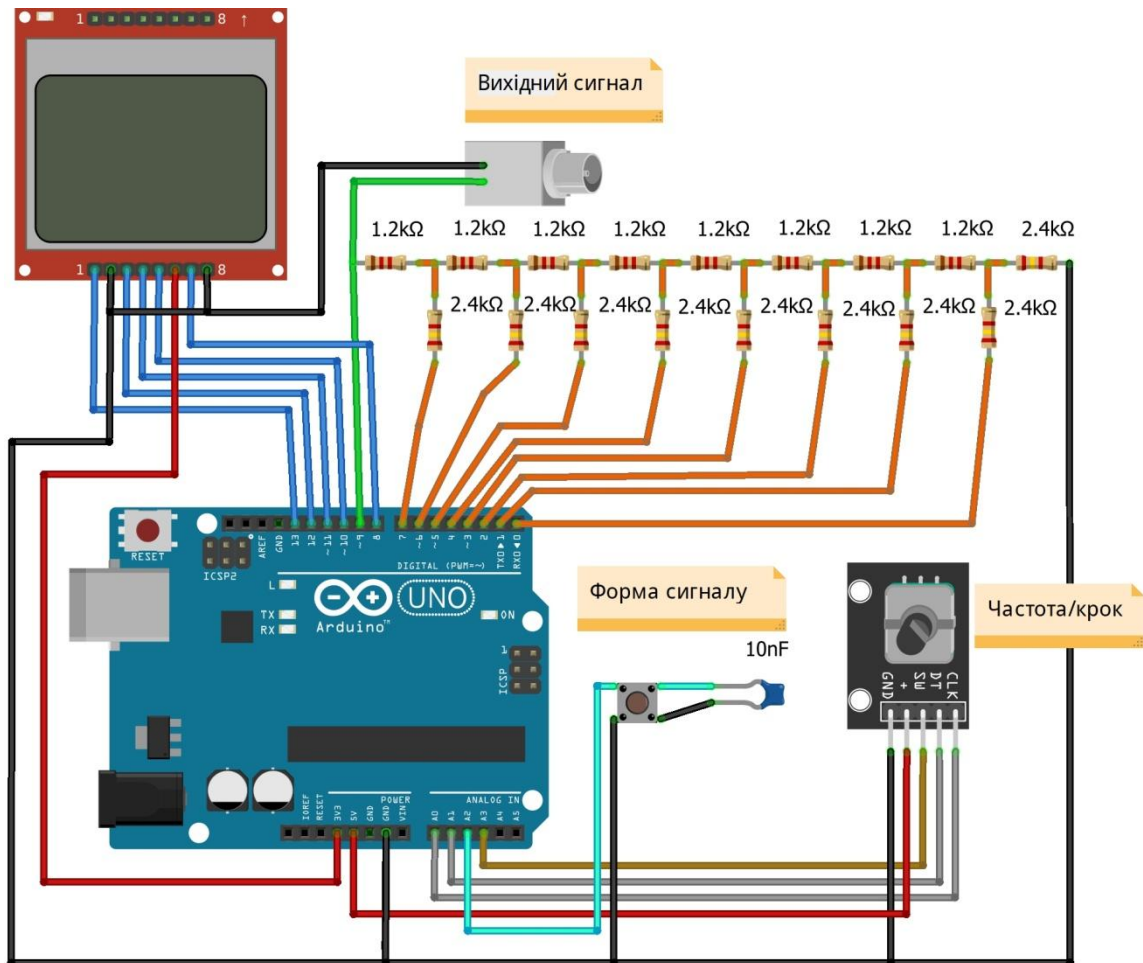


Рисунок 4.2 – Схема зборки генератора стандартних сигналів.

4.2 Розробка програмного коду

Генерація сигналів різної форми відбувається з використанням стандартних таблиць значення амплітуди, які зберігаються в пам'яті

мікроконтролера. За основу було взято алгоритм Джаспера (H. Jasper) Mini DDS [14].

Критична частина програми написана мовою Асемблера, оскільки потрібна велика швидкість роботи в основному циклі. Це і є «серце» синтезатора - декілька строк, що складають основний цикл. Оригінальний алгоритм був дещо змінено так, як він не дозволяв перемикаати форму вихідного сигналу. Для цього потрібно було змінити адресу початку таблиці значень та перезавантажити мікроконтролер.

Модифікація асемблерного коду дозволила припиняти DDS генерацію поточного сигналу в будь який момент.

Метод `Signal_Out` перевіряє, чи встановлений біт CPHA у регістрі SPCR, який встановлюється під час процедури обслуговування зовнішнього переривання (при натисканні кнопки зупинки). Алгоритм генерації DDS займає 10 циклів замість 9 без функції зупинки.

```
static inline void signalOUT(const uint8_t *signal){
    asm volatile( "eor r18, r18 ;r18<-0" "\n\t"
        "eor r19, r19 ;r19<-0" "\n\t"
        "1:" "\n\t"
        "add r18, %0 ;1 cycle" "\n\t"
        "adc r19, %1 ;1 cycle" "\n\t"
        "adc %A3, %2 ;1 cycle" "\n\t"
        "lpm ;3 cycles" "\n\t"
        "out %4, __tmp_reg__ ;1 cycle" "\n\t"
        "sbis %5, 7 ;1 cycle if no skip" "\n\t"
        "rjmp 1b ;2 cycles. Total 10 cycles" "\n\t"
        :
        : "r" (ad0), "r" (ad1), "r" (ad2), "e" (signal), "I"
        (_SFR_IO_ADDR(PORTD)), "I" (_SFR_IO_ADDR(PORTB))
        : "r18", "r19" );
}
```

Код функції генератора сигналів зчитує значення з довідкових таблиць, які зберігаються у флеш-пам'яті. Їх потрібно помістити в певні розділи, адреси яких починається з 0xxx00.

Функція на асемблері використовує не стандартний спосіб читання масиву, записаного через `PROGMEM` Для цього способу потрібно що б

розташування всіх таблиць-масивів було кратне адресі 0x100. В Arduino IDE немає штатних засобів що б покласти конкретний масив по конкретній адресі у флеш область [15], тому довелося обійти цей недолік не стандартним способом. Було створено спеціальний масив `musog_mass`, який займає все вільне місце аж до адреси 0x100, а після нього масиви з таблицями значень самі попадають точно на потрібні адреси. При зміні в коді або при використанні іншої версії Arduino IDE можливо буде потрібно зміна розміру «сміттевого масиву» таким чином, що б перший байт табличних масивів опинився точно за адресою 0x100. Перед завантаження коду в мікроконтролер потрібно перевірити адресу початку таблиць. Це можна зробити наступним чином: одразу після натискання кнопки "перевірити" відразу переглянути скомпільований hex файл, що створюється в тимчасовому каталозі. Приклад правильного розташування масивів наведено на рис. 4.3.

```

0      10      20      30      40      50      60
:10000000C9459050C9481050C9481050C94810580
:100010000C946F0A0C9481050C9481050C94810555
:100020000C9481050C9481050C9481050C94810538
:100030000C9481050C9481050C9481050C94810528
:100040000C94920F0C9481050C9481050C948105FD
:100050000C9481050C9481050C9481050C94810508
:100060000C9481050C9481054D08500853085608DE
:1000700059085C085F08000000000000000000054
:10008000000000000000000000000000000000070
:10009000000000000000000000000000000000060
:1000A000000000000000000000000000000000050
:1000B000000000000000000000000000000000040
:1000C000000000000000000000000000000000030
:1000D000000000000000000000000000000000020
:1000E000000000000000000000000000000000010
:1000F000000000000000000000000000000000000
:10010000FFFEFDFCFBFAF9F8F7F6F5F4F3F2F1F077
:10011000EFEEDCEBEAE9E8E7E6E5E4E3E2E1E067
:10012000DFDEDDCDBDAD9D8D7D6D5D4D3D2D1D057
:10013000CFCECDCBCBAC9C8C7C6C5C4C3C2C1C047
:10014000BFBEBDBCBBAB9B8B7B6B5B4B3B2B1B037
:10015000AFAEADACABAAA9A8A7A6A5A4A3A2A1A027
:100160009F9E9D9C9B9A9998979695949392919017
:100170008F8E8D8C8B8A8988878685848382818007
:100180007F7E7D7C7B7A79787776757473727170F7
:100190006F6E6D6C6B6A69686766656463626160E7
:1001A0005F5E5D5C5B5A59585756555453525150D7
:1001B0004F4E4D4C4B4A49484746454443424140C7
:1001C0003F3E3D3C3B3A39383736353433323130B7
:1001D0002F2E2D2C2B2A29282726252423222120A7
:1001E0001F1E1D1C1B1A1918171615141312111097
:1001F0000F0E0D0C0B0A0908070605040302010087
:10020000000102030405060708090A0B0C0D0E0F76

```

Допоміжний «нульовий» масив

Масив значень для «пили»

: 58 \$003A Text DOS ANSI Cyrillic (1251)

Рисунок 4.3 – Розташування табличних значень у пам'яті.

4.3 Технічні характеристики генератора

Характеристики при генерації за допомогою таймеру:

- генерація прямокутного сигналу 0..4 МГц;
- мінімальний крок регулювання частоти в діапазоні 0...2,8кГц - 1Гц;
- понад 2,8кГц мінімальний крок поступово збільшується;
- регулювання коефіцієнта заповнення (шпаруватості) 1..100%;
- у діапазоні 1Гц..80кГц регулювання проводиться з розв'язанням 1%;
- понад 80кГц дозвіл (крок) регулювання шпаруватості збільшується;

– при зміні робочої частоти в діапазоні 1Гц-80кГц обрана шпаруватість зберігається, а при зміні частоти понад 80кГц скидається на 50 % (значення за замовчуванням), але в режимі регулювання шпаруватості її можна знову змінювати.

Характеристики генерації сигналу методом DDS: максимальна частота обмежена до 100кГц.

С збільшенням частоти сильно падає роздільна здатність сигналу на високих частотах. А конкретно, у діапазонах:

- 0...6,25 кГц — розрядність знижується від 256 до 128 градацій;
- 6,25...12,5 кГц — розрядність знижується від 128 до 64 градацій;
- 12,5...25кГц — розрядність знижується від 64 до 32 градацій;
- 25кГц...50кГц — розрядність знижується від 32 до 16 градацій;
- 50кГц...100кГц — розрядність знижується від 16 до 8 градацій.

Крім цього зі збільшенням частоти зростає джитер, це особливо помітно на сигналах з різкими фронтами (прямокутник, пила).

Але на точно встановлених частотах 6,25кГц ; 12,5кГц ; 25кГц; 50кГц; 100 кГц джитер відсутній. Ці частоти можна використовувати для точних вимірювань. Інші частоти для більшості застосувань теж придатні.

Але якщо потрібний ідеальний сигнал — то тільки генерація таймером.

У всіх діапазонах DDS мінімальний крок регулювання частоти — 1Гц.

Керування пристроєм здійснюється за допомогою енкодера — регулювання вихідної частоти, кнопки енкодера — крок змінення частоти та додаткової кнопки — вибір форми вихідного сигналу.

Кнопка перемикання режимів послідовно переключає:

- (0) синус DDS;
- (1) трикутник DDS;
- (2) прямокутник DDS;
- (3) права пила DDS;
- (4) ліва пила DDS;
- (5) частота генерація таймером;

- (6) шпаруватість генерація таймером.

Кнопка на енкодері перемикає крок зміни частоти:

- 1 Гц;
- 10 Гц;
- 100 Гц;
- 1000 Гц;
- у режимі регулювання шпаруватості (6) натискання на кнопку перемикає енкодер у режим (5).

4.4 Тестування приладу

Перевірка роботи генератора стандартних сигналів здійснювалася за допомогою цифрового осцилографа (USB-приставки) «Oscill» [16]. Нижче наведені основні технічні характеристики осцилографа:

- смуга пропускання 15 МГц;
- розрядність АЦП 8 біт;
- частота дискретизації 32 Мвиб/с;
- у стробоскопічному режимі 800 Мвиб/с;
- мінімальна чутливість 10 В/діл;
- максимальна чутливість 20 мВ/діл;
- гранична вхідна напруга 250 В;
- обсяг ОЗП 2104 байт;
- інтерфейс зв'язку із ПК USB/UART;
- максимальна швидкість з'єднання із ПК 921кбод;
- гальванічна ізоляція від ПК 2500 В.

Усі вимірювання проводилися з відкритим входом осцилографа, щоб мінімізувати спотворення вихідного сигналу генератора.

Генерація сигналів методом DDS

На рис. 4.4 наведені осцилограми генеруемого синусоїдального

сигналу. Як видно з осцилограм, форма і якість генеруємих на максимальній частоті коливань майже не погіршується.

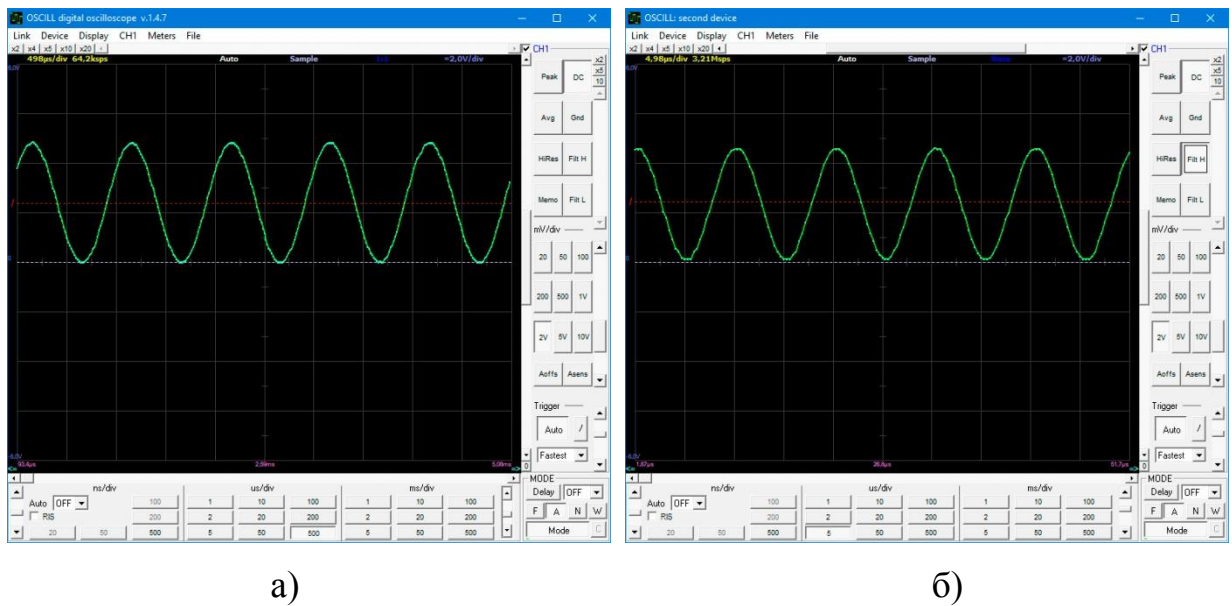
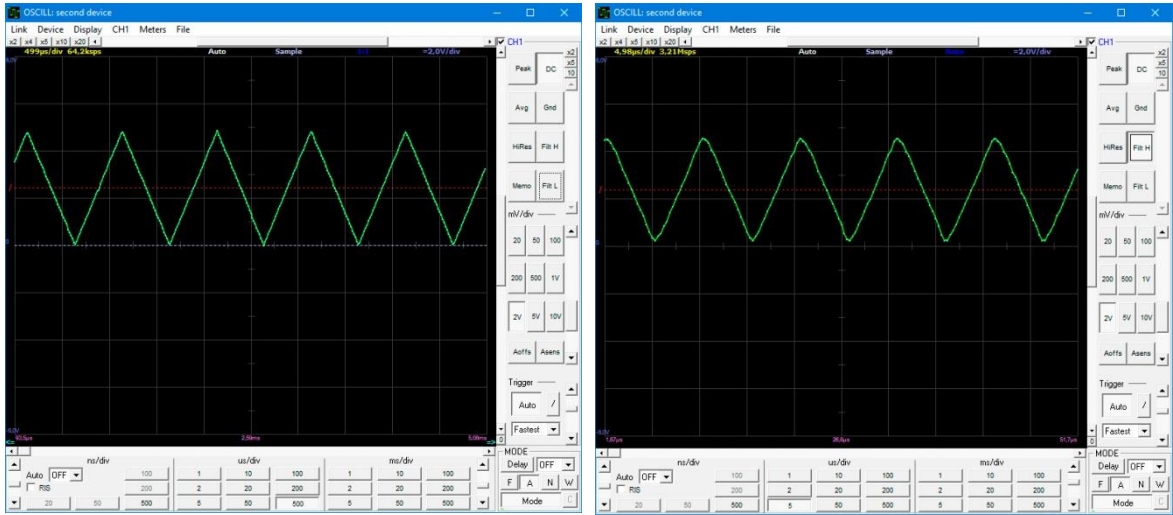


Рисунок 4.4 – Синусоїдальний сигнал, синтезований методом DDS:

а – частота 1 кГц, б – частота 100 кГц

На рис. 4.5. показано форма коливань трикутної форми на частотах 1 кГц та 100 кГц. На другій осцилограмі (100 кГц) видно, що відбувається незначне погіршення форми сигналу при мінімальній і максимальній амплітуді. Але в цілому форма сигналу залишається прийнятною для більшості використань.



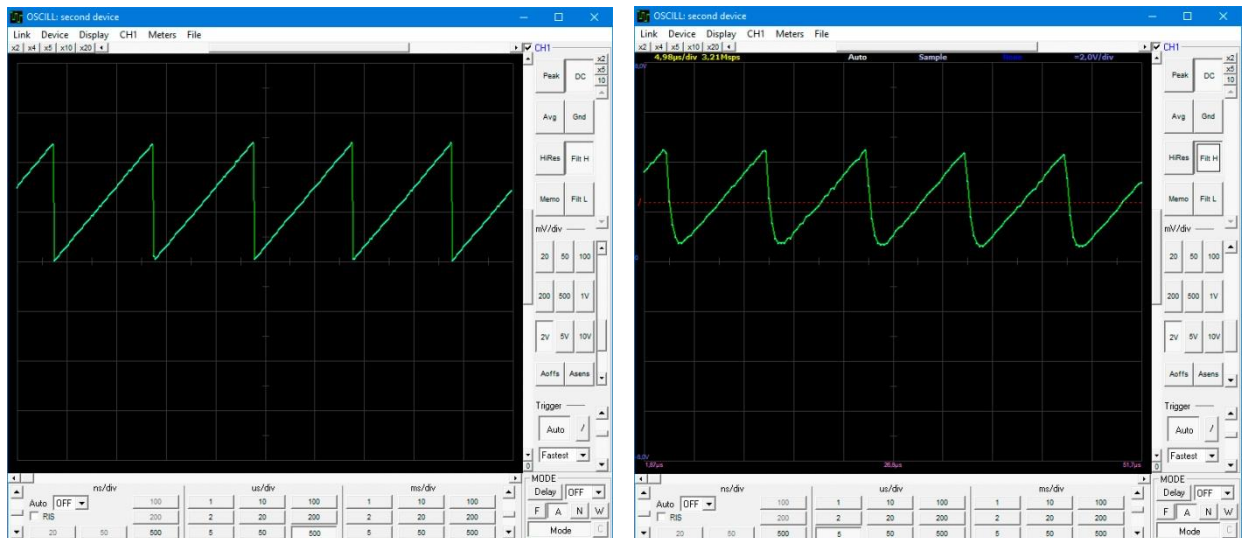
а)

б)

Рисунок 4.5 – Трикутний сигнал, синтезований методом DDS:

а – частота 1 кГц, б – частота 100 кГц

На рис. 4.6 та рис. 4.7 наведені форми осцилограм для пилоподібних сигналів прямої та зворотної форми на частотах 1 кГц та 100 кГц.



а)

б)

Рисунок 4.6 – «Прямий» пилоподібний сигнал: а – 1 кГц, б – 100 кГц

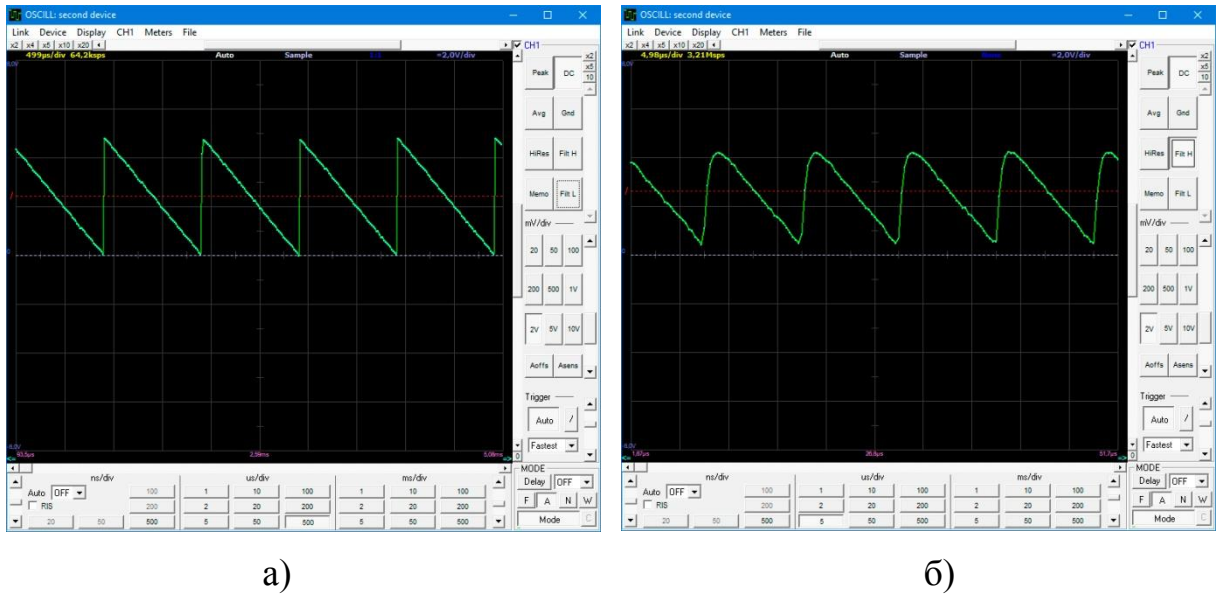


Рисунок 4.7 – «Зворотній» пилоподібний сигнал: а – 1 кГц, б – 100 кГц

Як видно з осцилограм, для обох видів пилоподібного сигналу на частоті 100 кГц характерне затягування фронтів та невелике спотворення форми сигналу одразу за фронтом. Такі спотворювання характерні при наявності в тракці проходження сигналу RC-ланцюгів. Оскільки в нашій схемі на виході генератора будь які конденсатори відсутні, а осцилограф підключено з відкритим входом, то спотворення форми сигналів можна пояснити внутрішніми паразитними ємностями порту мікроконтролера.

На рис. 4.8 показані форми прямокутного сигналу (меандр), генеруемого методом DDS. На частоті 100 кГц теж можна відзначити характерні для інтегруючих RC-ланцюгів спотворення форми сигналу.

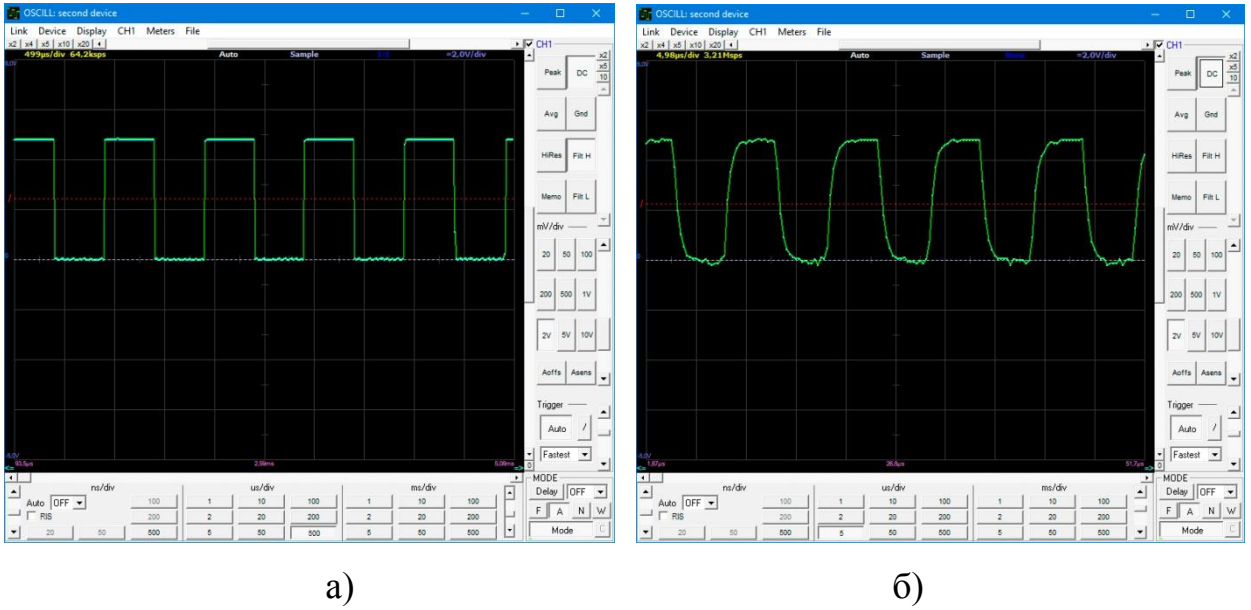


Рисунок 4.8 – Меандр, синтезований методом DDS:

а – частота 1 кГц, б – частота 100 кГц

Генерація сигналів за допомогою таймера

Для реалізації можливості регулювання коефіцієнту заповнення прямокутних імпульсів (шпаруватості) в алгоритмі програмного коду передбачено режим генерації за допомогою вбудованого тактового генератора. На рис. 4.9 наведені осцилограми прямокутних імпульсів частотою 1 кГц з коефіцієнтами заповнення 50%, 20% та 80%.

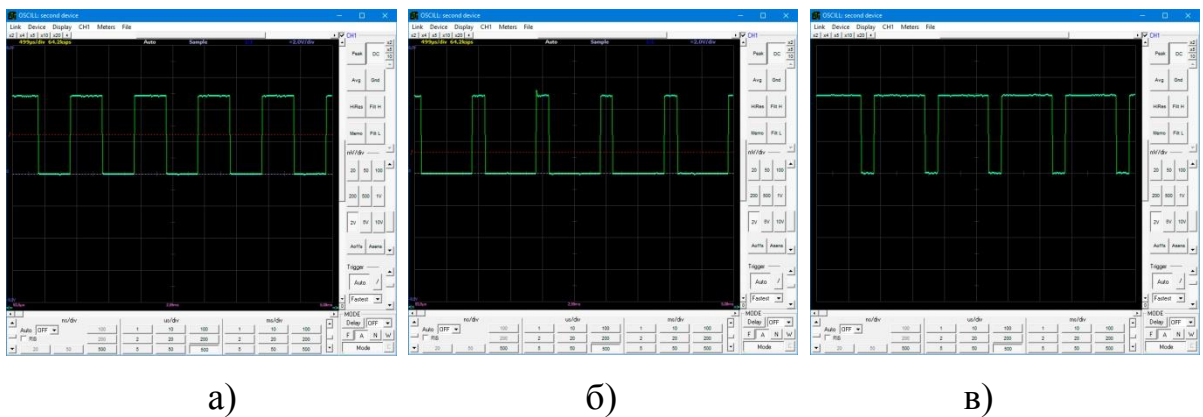


Рисунок 4.9 – Регулювання шпаруватості на частоті 1 кГц:

а – 50%, б – 20%, в – 80%

На рис. 4.10 показані такі самі налаштування для частоти 100 кГц.

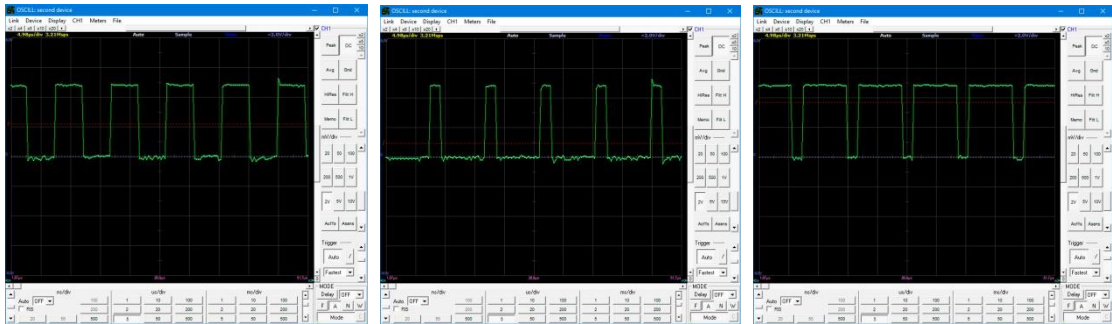
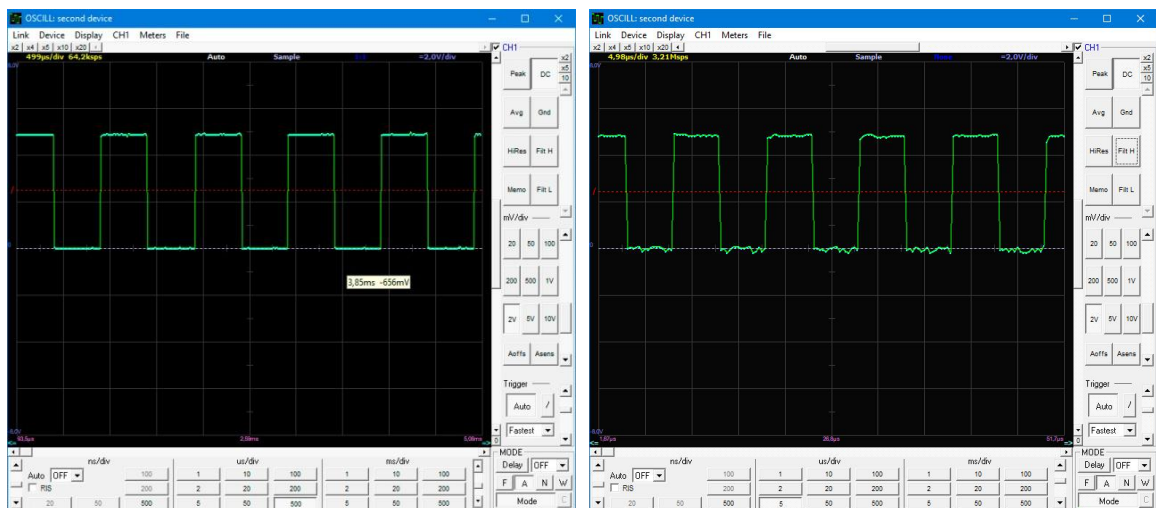


Рисунок 4.10 – Регулювання шпаруватості на частоті 100 кГц:

а – 50%, б – 20%, в – 80%

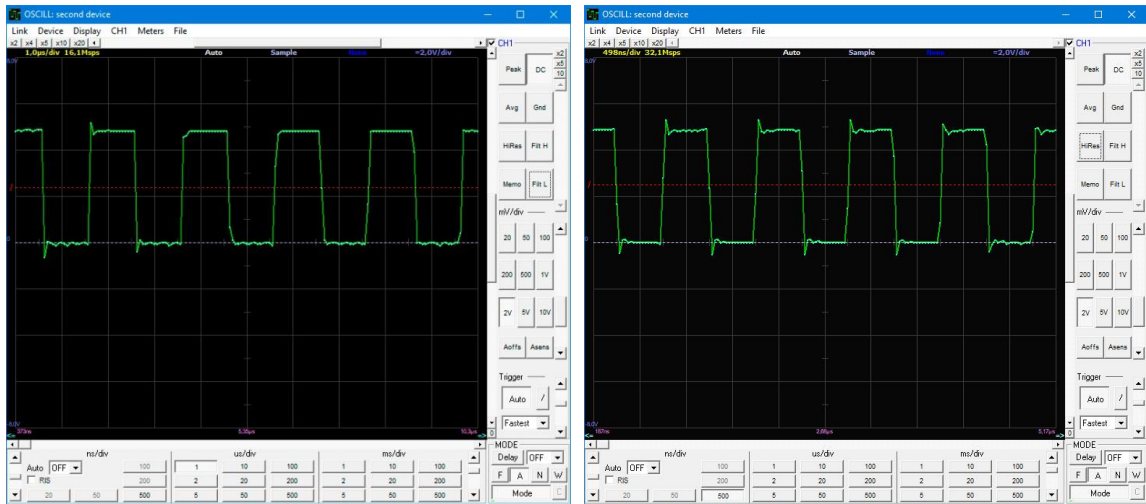
Порівнюючи осцилограми на рис. 4.9 та 4.10 можна зробити висновок, що регулювання шпаруватості відбувається однаково, а форма імпульсів відрізняється лише невеликим зтягуванням фронтів на частоті 100 кГц, що є нормальним явищем.

На рис. 4.11 показано як змінюється форма вихідних імпульсів (меандр) зі зростанням частоти.



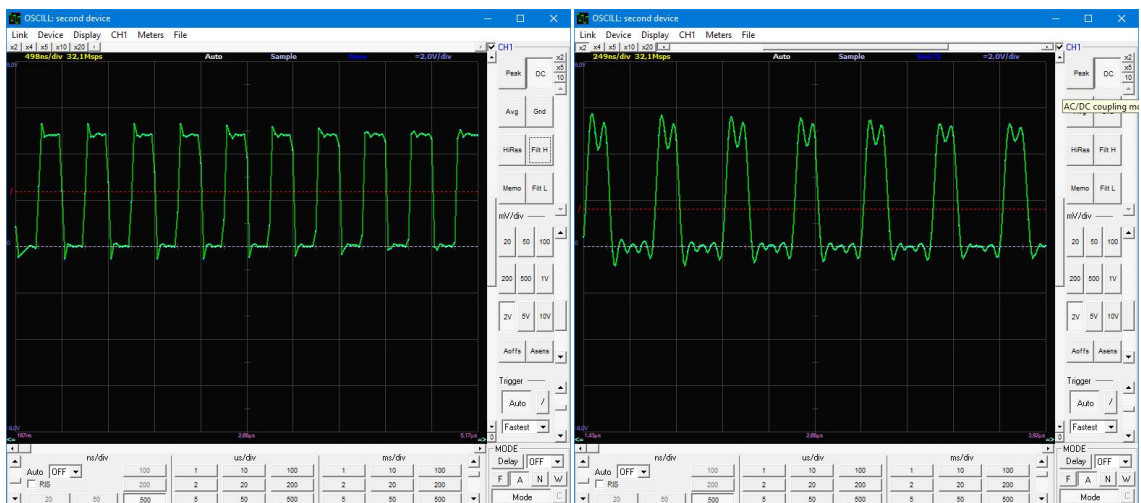
а)

б)



B)

Г)



Д)

Е)

Рисунок 4.11 – Залежність форми вихідних імпульсів від частоти при генерації по таймеру:

а – 1 кГц, б – 100 кГц, в – 500 кГц, г – 1 МГц, д – 2 МГц, е – 3 МГц

Як видно на рис. 4.11 (е) форма імпульсів на частоті 3 МГц є вже дуже спотвореною, але ще подібна до прямокутної. Гранична частота при генерації по таймеру прямокутних імпульсів обмежена до 4 МГц, але при цій частоти форма вихідного сигналу має вже незадовільну форму, що видно на рис. 4.12.

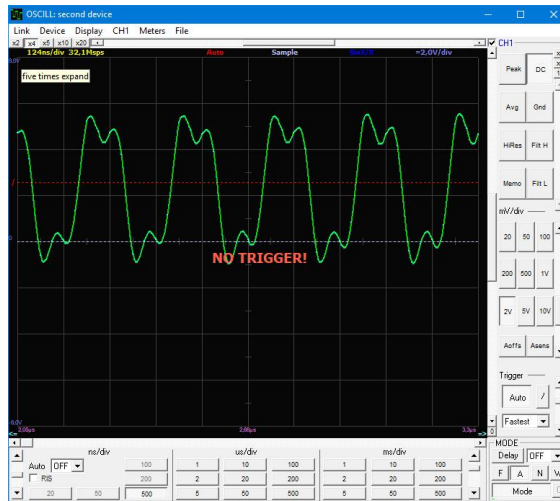


Рисунок 4.12 – Форма вихідного сигналу прямокутної форми на частоті 4 МГц.

Проаналізувавши наведені осцилограми можна зробити наступні висновки:

1. При прямому цифровому синтезі (DDS) гранична вихідна частота для сигналів прямокутної, трикутної та пилоподібної форми становить близько 100 кГц. При ці частоті вже відбувається спотворення сигналів та затягування фронтів. Це можна пояснити не ідеальністю електронних ключів на виходах мікроконтролера — не нульовий опір відчиненого ключа, паразитна ємність та ін. А оскільки при для синтезу використовується відразу декілька портів, то вплив цих чинників збільшується.
2. Частота синтезу синусоїдального сигналу може перевищувати межу 100 кГц и обмежується бистродією мікроконтролера, тобто кількістю відліків на одиницю часу.
3. При генерації прямокутного сигналу за допомогою таймеру використовується тільки один порт і, відповідно, вплив паразитної ємності і опору мінімізується, що дає можливість збільшити вихідну частоту до кількох МГц.

ВИСНОВКИ

В процесі виконання кваліфікаційної роботи були вивчені та розглянуті методи цифрового синтезу електричних сигналів, в тому числі і прямий цифровий синтез сигналів на основі табличних значень. Було проаналізовано можливість використання платформи Arduino для створення генератора сигналів стандартних функцій.

На основі отриманих даних була підібрана технічна та теоретична база для реалізації проекту. На базі плати Arduino UNO та резистивного ЦАП R-2R створено прототип генератора стандартних сигналів.

Розроблено програмне забезпечення для синтезу сигналів різноманітної форми. Генерація таких сигналів, як синусоїдальний, трикутний, пилоподібний та прямокутний (меандр), здійснюється за допомогою технології прямого цифрового синтезу (DDS) на основі табличних значень функцій. Такий метод дає можливість встановлювати вихідну частоту з великою точністю - 1 Гц, але обмежує верхню межу частот. Це обумовлено бистродією мікроконтролера. Не зважаючи на те, що частина програмного коду для підвищення швидкості виконання була написана мовою Assembler, не вдалося отримати сигнали прийнятної якості на частотах вище 100 кГц. Тому для розширення частотного діапазону прямокутних сигналів та можливості регулювання шпаруватості, в генератор було добавлено ще два режими роботи, де генерування сигналів відбувається за допомогою вбудованого в мікроконтролер таймеру. Це на дало можливість підвищити верхню частоту вихідних імпульсів майже до 4 МГц.

Для збільшення навантажувальної здатності генератора бажано на його виході встановити високоякісний операційний підсилювач. Але це потребує додаткового блоку живлення (+/-15 В).

Для отримання більш високих частот вихідних сигналів необхідно використати потужніші мікроконтролери, наприклад STM32.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Rohde & Schwarz. Модуляція і формування сигналів з допомогою генераторів сигналів компанії R&S. [Електроний ресурс]. – Режим доступу: https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_application/application_notes/1ma225/1MA225_1rus_Modulation_Signal_Generation.pdf
2. Tektronix. Генераторы сигналов от А до Я. Учебное пособие. [Електроний ресурс]. – Режим доступу: <https://download.tek.com/document/76U-16672-5.pdf>
3. Дьяконов В. П. Генерация и генераторы сигналов / В. П. Дьяконов. — М. : ДМК Пресс, 2009. — 384 с.
4. Brendan Cronin. DDS Devices Generate High-Quality Waveforms Simply, Efficiently, and Flexibly [Електроний ресурс]. – Режим доступу: <https://www.analog.com/en/analog-dialogue/articles/dds-generates-high-quality-waveforms-efficiently.html>
5. Шауэрман Александр А. Исследование способов генерации сигналов. Методическое указание к лабораторной работе. [Електроний ресурс]. – Режим доступу: <http://www.labfor.ru/guidance/fpga-dsp/dds>
6. Аппаратная платформа Arduino [Електроний ресурс]. URL: <https://www.arduino.ua/>
7. Arduino Uno Rev3 (ATmega16U2) : [Електроний ресурс]. – Режим доступу: <https://arduino.ua/prod676-arduino-uno-rev3>
8. Модуль Энкодер : [Електроний ресурс]. – Режим доступу: <https://arduino.ua/prod1223-modul-enkoder>
9. Nokia 5110 LCD модуль [Електроний ресурс]. – Режим доступу: <https://arduino.ua/prod407-nokia-5110-lcd-modul>

10. ЦАП на основе резистивной матрицы R-2R и его реализация на микроконтроллере. [Электроний ресурс]. – Режим доступа: https://radioham.ru/dac_r-2r/
11. Юрий Карлов. Цап R2R – нюансы. [Электроний ресурс]. – Режим доступа: <https://marsohod.org/11-blog/111-r2rdac>
12. Соммер У. Программирование микроконтроллерных плат Arduino/Freduino. - СПб.: БХВ-Петербург, 2012. -256 с., ил. - (Электроника)
13. Fritzing. [Электроний ресурс]. – Режим доступа: <https://fritzing.org/>
14. Н. Jasper Mini DDS. [Электроний ресурс]. – Режим доступа: <http://www.radanpro.com/Radan2400/mikrokontroleri/Jesper%27s%20AVR%20pages%20-%20MiniDDS.htm>
15. Дж. Блум «Изучаем Arduino. Инструменты и методы технического волшебства». – М.: Эком, 2015. – 336 с.
16. Универсальный любительский осциллограф-приставка к ПК. [Электроний ресурс]. – Режим доступа: <http://oscill.ru/rus/oscilloscopes.html>

ДОДАТОК А

Код програми для мікроконтролера

```

/* Генератор з регульованою частотою й шпаруватістю
 * Підключення енкодера A0,A1 (in)
 * Кнопка енкодера A3 (in)
 * Кнопка "режим" A2 (in)
 * Підсвітка 8 (out) (контакт ВL дисплея)
 * Дисплей nokia 5110 - 13,12,11,10 (out)
 * Вихід генератора 0..7 + 9 (out)
 */

const PROGMEM uint8_t  sinewave[] = // масив синусоїдального
сигналу
{
0x80,0x83,0x86,0x89,0x8c,0x8f,0x92,0x95,0x98,0x9c,0x9f,0xa2,0xa5
,0xa8,0xab,0xae,
0xb0,0xb3,0xb6,0xb9,0xbc,0xbf,0xc1,0xc4,0xc7,0xc9,0xcc,0xce,0xd1
,0xd3,0xd5,0xd8,
0xda,0xdc,0xde,0xe0,0xe2,0xe4,0xe6,0xe8,0xea,0xec,0xed,0xef,0xf0
,0xf2,0xf3,0xf5,
0xf6,0xf7,0xf8,0xf9,0xfa,0xfb,0xfc,0xfc,0xfd,0xfe,0xfe,0xff,0xff
,0xff,0xff,0xff,
0xff,0xff,0xff,0xff,0xff,0xff,0xfe,0xfe,0xfd,0xfc,0xfc,0xfb,0xfa
,0xf9,0xf8,0xf7,
0xf6,0xf5,0xf3,0xf2,0xf0,0xef,0xed,0xec,0xea,0xe8,0xe6,0xe4,0xe2
,0xe0,0xde,0xdc,
0xda,0xd8,0xd5,0xd3,0xd1,0xce,0xcc,0xc9,0xc7,0xc4,0xc1,0xbf,0xbc
,0xb9,0xb6,0xb3,
0xb0,0xae,0xab,0xa8,0xa5,0xa2,0x9f,0x9c,0x98,0x95,0x92,0x8f,0x8c
,0x89,0x86,0x83,
0x80,0x7c,0x79,0x76,0x73,0x70,0x6d,0x6a,0x67,0x63,0x60,0x5d,0x5a
,0x57,0x54,0x51,
0x4f,0x4c,0x49,0x46,0x43,0x40,0x3e,0x3b,0x38,0x36,0x33,0x31,0x2e
,0x2c,0x2a,0x27,
0x25,0x23,0x21,0x1f,0x1d,0x1b,0x19,0x17,0x15,0x13,0x12,0x10,0x0f
,0x0d,0x0c,0x0a,
0x09,0x08,0x07,0x06,0x05,0x04,0x03,0x03,0x02,0x01,0x01,0x00,0x00
,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x01,0x02,0x03,0x03,0x04,0x05
,0x06,0x07,0x08,
0x09,0x0a,0x0c,0x0d,0x0f,0x10,0x12,0x13,0x15,0x17,0x19,0x1b,0x1d
,0x1f,0x21,0x23,
0x25,0x27,0x2a,0x2c,0x2e,0x31,0x33,0x36,0x38,0x3b,0x3e,0x40,0x43
,0x46,0x49,0x4c,
0x4f,0x51,0x54,0x57,0x5a,0x5d,0x60,0x63,0x67,0x6a,0x6d,0x70,0x73
,0x76,0x79,0x7c
};

```



```

0xe0,0xe2,0xe4,0xe6,0xe8,0xea,0xec,0xee,0xf0,0xf2,0xf4,0xf6,0xf8
,0xfa,0xfc,0xfe,
0xff,0xfd,0xfb,0xf9,0xf7,0xf5,0xf3,0xf1,0xef,0xef,0xeb,0xe9,0xe7
,0xe5,0xe3,0xe1,
0xdf,0xdd,0xdb,0xd9,0xd7,0xd5,0xd3,0xd1,0xcf,0xcf,0xcb,0xc9,0xc7
,0xc5,0xc3,0xc1,
0xbf,0xbd,0xbb,0xb9,0xb7,0xb5,0xb3,0xb1,0xaf,0xaf,0xab,0xa9,0xa7
,0xa5,0xa3,0xa1,
0x9f,0x9d,0x9b,0x99,0x97,0x95,0x93,0x91,0x8f,0x8f,0x8b,0x89,0x87
,0x85,0x83,0x81,
0x7f,0x7d,0x7b,0x79,0x77,0x75,0x73,0x71,0x6f,0x6f,0x6b,0x69,0x67
,0x65,0x63,0x61,
0x5f,0x5d,0x5b,0x59,0x57,0x55,0x53,0x51,0x4f,0x4f,0x4b,0x49,0x47
,0x45,0x43,0x41,
0x3f,0x3d,0x3b,0x39,0x37,0x35,0x33,0x31,0x2f,0x2f,0x2b,0x29,0x27
,0x25,0x23,0x21,
0x1f,0x1d,0x1b,0x19,0x17,0x15,0x13,0x11,0x0f,0x0f,0x0b,0x09,0x07
,0x05,0x03,0x01
};

```

```

const PROGMEM uint8_t sawtoothwave[] = //масив пилоподібного
сигналу 1
{
0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0a,0x0b,0x0c
,0x0d,0x0e,0x0f,
0x10,0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18,0x19,0x1a,0x1b,0x1c
,0x1d,0x1e,0x1f,
0x20,0x21,0x22,0x23,0x24,0x25,0x26,0x27,0x28,0x29,0x2a,0x2b,0x2c
,0x2d,0x2e,0x2f,
0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39,0x3a,0x3b,0x3c
,0x3d,0x3e,0x3f,
0x40,0x41,0x42,0x43,0x44,0x45,0x46,0x47,0x48,0x49,0x4a,0x4b,0x4c
,0x4d,0x4e,0x4f,
0x50,0x51,0x52,0x53,0x54,0x55,0x56,0x57,0x58,0x59,0x5a,0x5b,0x5c
,0x5d,0x5e,0x5f,
0x60,0x61,0x62,0x63,0x64,0x65,0x66,0x67,0x68,0x69,0x6a,0x6b,0x6c
,0x6d,0x6e,0x6f,
0x70,0x71,0x72,0x73,0x74,0x75,0x76,0x77,0x78,0x79,0x7a,0x7b,0x7c
,0x7d,0x7e,0x7f,
0x80,0x81,0x82,0x83,0x84,0x85,0x86,0x87,0x88,0x89,0x8a,0x8b,0x8c
,0x8d,0x8e,0x8f,
0x90,0x91,0x92,0x93,0x94,0x95,0x96,0x97,0x98,0x99,0x9a,0x9b,0x9c
,0x9d,0x9e,0x9f,
0xa0,0xa1,0xa2,0xa3,0xa4,0xa5,0xa6,0xa7,0xa8,0xa9,0xaa,0xab,0xac
,0xad,0xae,0xaf,
0xb0,0xb1,0xb2,0xb3,0xb4,0xb5,0xb6,0xb7,0xb8,0xb9,0xba,0xbb,0xbc
,0xbd,0xbe,0xbf,
0xc0,0xc1,0xc2,0xc3,0xc4,0xc5,0xc6,0xc7,0xc8,0xc9,0xca,0xcb,0xcc
,0xcd,0xce,0xcf,
0xd0,0xd1,0xd2,0xd3,0xd4,0xd5,0xd6,0xd7,0xd8,0xd9,0xda,0xdb,0xdc
,0xdd,0xde,0xdf,
0xe0,0xe1,0xe2,0xe3,0xe4,0xe5,0xe6,0xe7,0xe8,0xe9,0xea,0xeb,0xec
,0xed,0xee,0xef,

```



```

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,
};

```

```

volatile int32_t freq=1000;// частота за замовчуванням
volatile uint16_t shag=100; //крок енодера за замовчуванням
volatile uint8_t regim=5; //режим генератора за замовчуванням
volatile uint8_t monitor_flag; // прапор для виводу на дисплей
volatile uint8_t ad2,ad1,ad0; //службові байти для функції на
мові асемблера
volatile uint8_t duty=50; // шпаруватість
volatile uint32_t icr=0; // змінна для керування регістром
порівняння таймерал

```

```

#include <LCD5110_SSVS.h>
extern uint8_t SmallFont[];
extern uint8_t MediumNumbers[];
LCD5110 lcd(10,11,12,13);

```

```

// DDS алгоритм
static inline void signalOUT(const uint8_t *signal){
    asm volatile( "eor r18, r18 ;r18<-0" "\n\t"
        "eor r19, r19 ;r19<-0" "\n\t"
        "1:" "\n\t"
        "add r18, %0 ;1 cycle" "\n\t"
        "adc r19, %1 ;1 cycle" "\n\t"
        "adc %A3, %2 ;1 cycle" "\n\t"
        "lpm ;3 cycles" "\n\t"
        "out %4, __tmp_reg__ ;1 cycle" "\n\t"
        "sbis %5, 7 ;1 cycle if no skip" "\n\t"
        "rjmp 1b ;2 cycles. Total 10 cycles" "\n\t"
        :
        : "r" (ad0), "r" (ad1), "r" (ad2), "e" (signal), "I"
        (_SFR_IO_ADDR(PORTD)), "I" (_SFR_IO_ADDR(PORTB))
        : "r18", "r19" );
}

```

```

void setup(){
lcd.InitLCD();
for (int n=0; n <sizeof(musor_mass); n++ ) {PORTD=musor_mass[n];
}
PORTC|=(1<<PC2)| (1<<PC3); // «підтяжка» кнопок на A2 A3
TCCR0B=0; TCCR1A=0; TCCR1B=0;

```

```

    DDRB|=1<<PB0; // 8 пін -включення підсвічування дисплея =
вихід
    PCICR=(1<<PCIE1); // ввімкнути переривання PCINT
    PCMSK1=(1<<PCINT9)|(1<<PCINT10)|(1<<PCINT11); // вибір пінів
переривання (A1,A2,A3)
    check_regim();
    } //end setup

void up_down(boolean x){ // керування регулюваннями

if (TCCR1B==17 && ICR1<2800 && regim==5){
if(x) { if (icr<1000 && shag > 100) shag=100;
        if (icr<100 && shag > 10) shag=10;
        if (icr<10 && shag > 1) shag=1;
        icr-=shag;
        if (icr<2) icr=2;    }
    else { if (icr > 1800 && shag >100) shag =100;
        icr+=shag ;    }
        return;
    }
if (regim==6){

    if (ICR1>100){

        if (x){if(duty<100) {duty++; } }

        if (!x){ if(duty>0) {duty--; }}
        }
    else{
        if (x){if(OCR1A<ICR1) {OCR1A++; } }
        else {if(OCR1A>0) {OCR1A--; } }
        if(OCR1A>ICR1) OCR1A=ICR1-1;
        duty=(uint32_t)100*OCR1A/ICR1;
    }
    return;
}

x? freq+=shag : freq-=shag ;
if (freq < 1) freq=1;
}

// переривання від кнопок і енодера
ISR (PCINT1_vect){
    PCMSK1=0; PCICR=0;
    PORTD=0;
    // якщо натиснута кнопка енодера
    if ((PINC&(1<<3))==0){
        while ((PINC&(1<<3))==0); // зачекати до тих пір, коли
кнопку відпустять

```

```

// блок вкл / вимк підсвітки дисплея (8 пін)
    if ((PINC&(1<<2))==0){ // якщо після цього натиснута
кнопка режимів,
        PINB|=1<<PB0; // змінювати стан 8 піна
        while ((PINC&(1<<2))==0); // тепер почекати поки
відпустять кнопку режиму
        PCMSK1=(1<<PCINT9) | (1<<PCINT10) | (1<<PCINT11);
PCICR=(1<<PCIE1); return;
    }
    if (regim==6) { regim=5; check_regim();
PCMSK1=(1<<PCINT9) | (1<<PCINT10) | (1<<PCINT11); PCICR=(1<<PCIE1);
return; }
    switch (shag){
        case 1: shag=10; break;
        case 10: shag=100; break;
        case 100: shag=1000; break;
        case 1000: shag=1;
    }
    check_regim();
    PCMSK1=(1<<PCINT9) | (1<<PCINT10) | (1<<PCINT11);
PCICR=(1<<PCIE1); return;
}
if ((PINC&(1<<2))==0){ // якщо натиснута кнопка режимів
    PORTD=0;
    while ((PINC&(1<<2))==0);
    regim++; if (regim==7) regim=0;
    check_regim();
PCMSK1=(1<<PCINT9) | (1<<PCINT10) | (1<<PCINT11); PCICR=(1<<PCIE1);
return;
}

// якщо кнопки не натискалися значить крутили енкодер:
    up_down( (PINC&1)^((PINC&2)>>1) ); // відправити
зчитане з енкодера в іншу функцію
    // якщо при обертанні рахунок йде не в потрібний бік, то
вставити символ '!' up_down(!
    check_regim();
    PCMSK1=(1<<PCINT9) | (1<<PCINT10) | (1<<PCINT11);
PCICR=(1<<PCIE1); // ввімкнути переривання

} // кінець функції обробки переривань від кнопок
PCINT1_vect

void pwm_gen(){ // налаштування таймера1
    uint16_t divider=1;
    icr = (16000000ul / freq / 2 / divider);
    byte shifts[] = {3,3,2,2};
    for(byte i = 0; i < 4; i++){
        if (icr > 65536) {
            divider <=< shifts[i];
            icr = 16000000ul / freq / 2 / divider;
        }
    }
}

```

```

        else {
            TCCR1B = (i+1)|(1<<WGM13); break;
        }
    }
    ICR1=icr-1;
    set_duty();
} //end pwm_gen

void loop() {
    if (monitor_flag) { monitor_flag=0; monitor_out(); }
    if (regim <5){ PORTB&= ~(1<<7);
    DDRD=0xFF;//set D port as output
    uint32_t temp=(float)freq /0.095367431640625;
    ad2=temp>>16; ad1=temp>>8; ad0=temp;
    switch (regim){
        case 0: signalOUT(sinewave); break;
        case 1: signalOUT(trianglewave); break;
        case 2: signalOUT(squarewave); break;
        case 3: signalOUT(sawtoothwave); break;
        case 4: signalOUT(rewsawtoothwave);
    } //end switch
    DDRD=0; PORTD=0;

    } //end if (regim<5)

}

void monitor_out(){
String dutystr,stepstr;
dutystr= String("Duty="+ String(duty)+ "%");
stepstr=String("Step= "+String(shag));
lcd.clrScr();
// Виведення першого рядка
lcd.setFont(SmallFont);
if (freq <10000) {lcd.print("Frequency, Hz",LEFT,0 ); }
if (freq >=10000) {lcd.print("Frequency, kHz",LEFT,0 ); }
// Виведення другого рядка
lcd.setFont(MediumNumbers);
if (freq <10000) { lcd.printNumI(freq, CENTER, 8); }
if (freq >=10000u && freq < 1000000ul ) { lcd.printNumF(
((float)freq/1000),3 ,CENTER, 8); }
if (freq >=1000000ul ) { lcd.printNumF( ((float)freq/1000),2
,CENTER, 8); }
// Виведення третього рядка
lcd.setFont(SmallFont);
switch(regim){
    case 0: lcd.print("Sinus DDS",CENTER, 32); break;
    case 1: lcd.print("Triangle DDS",CENTER, 32); break;
    case 2: lcd.print("Meandr DDS",CENTER, 32); break;
    case 3: lcd.print("Pila1 DDS",CENTER, 32); break;
    case 4: lcd.print("Pila2 DDS",CENTER, 32); break;
}
}

```

```

        case 5: lcd.print("PWM Mode",CENTER,32); break;
        case 6: lcd.print("Duty Mode", CENTER, 32);
    }
// Виведення четвертого рядка
    if (regim==6) lcd.print(dutystr, CENTER, 40);
        else lcd.print(stepstr, CENTER, 40);
} //end monitor out

void set_duty(){
//float ocr,delta_ocr;
if (regim==6&&ICR1<100) return;
if (regim==5 && ICR1<100){
    OCR1A=ICR1/2; duty=50;
    return;
}
static uint16_t ocr;
ocr=(uint32_t)ICR1*duty/100;
OCR1A=ocr;
}

void check_regim(){//перевірка і встановлення режимів генератора
if (regim <5){ if (freq > 100000) freq=100000;
    TCCR1B=0;
    TCCR1A=0; DDRB&=~(1<<DDB1); // 9 пін arduino
встановлюється Z-режим
} // end if regim <5

    if (regim > 4) {
        TCCR1A=1<<COM1A1; DDRB|=1<<DDB1; // 9 пін arduino
встановлюється в режим виходу (pwm out)
        if (TCCR1B==17 && ICR1<2800){
            ICR1=icr; freq= (float) 8000000UL/ICR1;
            set_duty();
        }
        else { pwm_gen(); }
    }
    PORTB|= 1<<7;
monitor_flag=1;
}

```