

Лабораторна робота №7

Дослідження ШІМ сигналу

Тема: Дослідження особливостей сигналу широтно-імпульсної модуляції

Мета: Дослідити особливості ШІМ сигналу та методи керування за допомогою нього.

У радіоелектроніці застосовуються сигнали двох видів: аналогові й дискретні (рис. 7.1). Аналогові сигнали мають безперервний характер і в заданому діапазоні рівнів можуть приймати будь-які значення напруг і струмів. Дискретні сигнали мають переривчастий характер, як за часом, так і за рівнем.

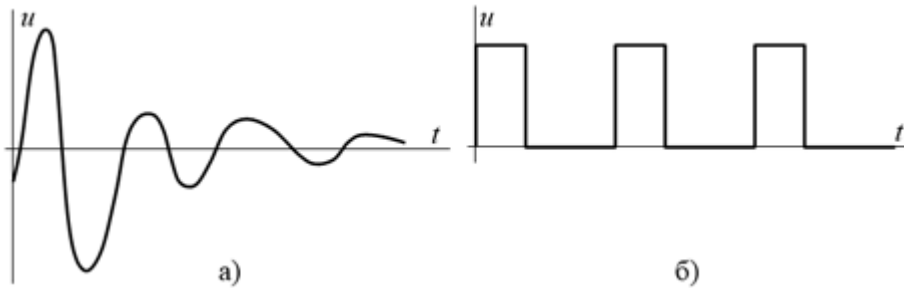


Рис. 7.1. Вид сигналу: а) аналоговий; б) дискретний.

Пристрою, що використовують дискретні сигнали працюю в імпульсному режимі, при якому чергуються робочі моменти й паузи. Під електричним імпульсом розуміють короткочасну зміну струму й (або) напруги, яка може відбуватися за різними законами, що визначає форму імпульсів (рис. 7.2).

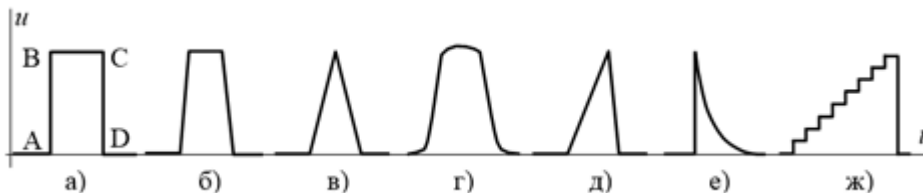


Рис. 7.2. Форма ідеалізованих імпульсних сигналів: а) прямокутний; б) трапецеїдальний; в) трикутний; г) дзвіноподібний; д) пилоподібний; е) експоненціальний; ж) ступеневий.

Прийнято розрізняти наступні ділянки імпульсу (рис. 7.2, а): фронт (АВ), вершина (ВС), зріз (CD), основа (AD). Фронт відповідає швидкому наростанню сигналу до максимального значення; вершина – порівняно повільній зміні сигналу протягом деякого проміжку часу; зріз – швидкому убаванню імпульсу.

Реальний прямокутний імпульс відрізняється від ідеалізованого внаслідок перехідних процесів, що відбуваються в ланцюгах. Параметрами імпульсу (рис. 7.3) є:

- амплітуда імпульсу U_m , тобто найбільша напруга імпульсного сигналу;
- тривалість імпульсу t_i , що характеризує тривалість імпульсу в часі;
- тривалість фронту t_f і тривалість зрізу імпульсу t_c , характеризують відповідно час наростання й спаду імпульсу;

– спад вершини імпульсу ΔU - відображає зменшення напруги на плоскій частині імпульсу.

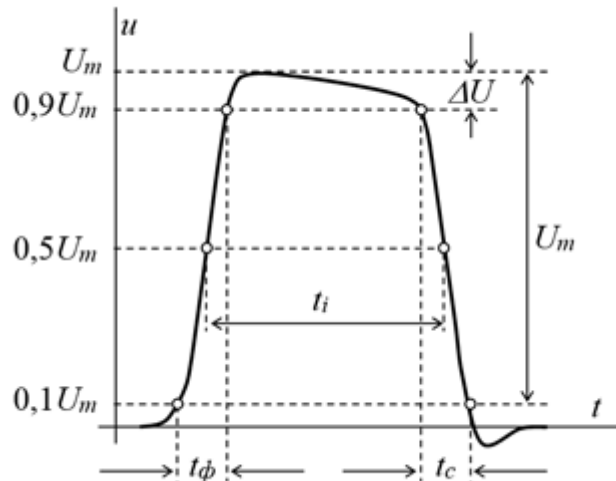


Рис. 7.3. Форма реального прямокутного імпульсу і його параметри.

Параметрами послідовності імпульсів (рис. 7.4) є:

- період повторення імпульсів T , інтервал часу між відповідними точками (наприклад, між початками) двох сусідніх імпульсів;
- частота повторення імпульсів f - величина, зворотна періоду повторення $f = 1/T$;
- тривалість паузи t_p - інтервал часу між закінченням одного й початком наступного імпульсу: $t_p = T - t_i$;
- коефіцієнт заповнення γ - відношення тривалості імпульсів до періоду їх проходження: $\gamma = t_i/T$;
- скважність імпульсів q , величина зворотна коефіцієнту заповнення: $q = T/t_i = 1/\gamma$.

Скважність визначає відношення пікової потужності імпульсу до його середньої потужності. Зміна скважності періодичної послідовності імпульсів дозволяє регулювати середню величину напруги, що подається на навантаження.

В імпульсній схемотехніці часто використовується періодичний сигнал прямокутної форми із скважністю, що дорівнює двом – меандр, у цьому випадку тривалість імпульсу дорівнює тривалості паузи.

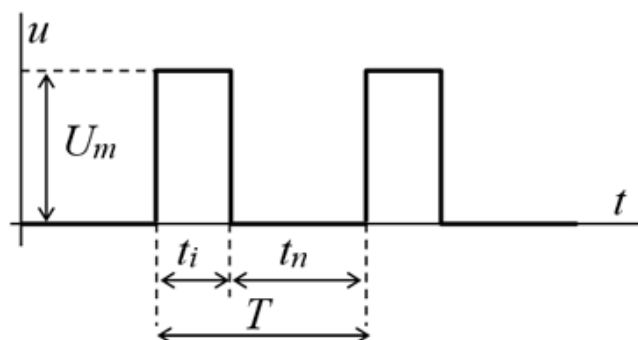


Рис. 7.4. Періодична послідовність імпульсів прямокутної форми.

Широтно-імпульсна модуляція (ШІМ — англ. pulse-width modulation, PWM), або модуляція за тривалістю імпульсів (англ. pulse-duration modulation, PDM) — процес керування шириною (тривалістю) високочастотних імпульсів за законом, який задає низькочастотний сигнал. В електроніці це може бути керування середнім значенням вихідної напруги шляхом зміни тривалості замкнутого стану електронного (електромеханічного) ключа, наприклад, у схемі ключового стабілізатора напруги.

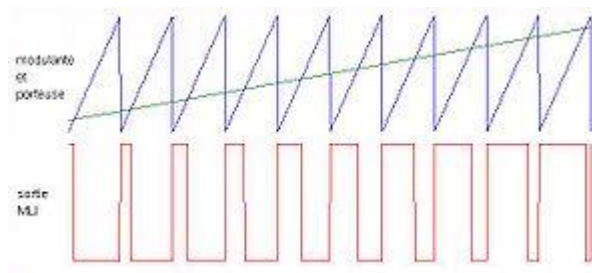


Рис.7.5. Зелений — вхідний аналоговий сигнал, червоний — вихідний імпульсний

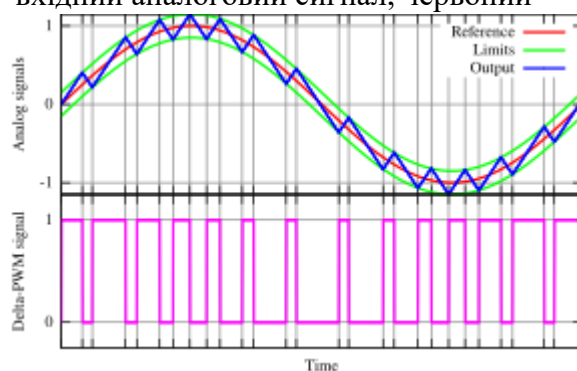


Рис.7.6. Моделювання в ШІМ вихідного синусоїдального сигналу

Аналогова ШІМ

ШІМ-сигнал генерується аналоговим компаратором, на один вхід якого подається опорний сигнал значно більшої частоти, ніж модулюючий у вигляді «трикутника» або «пили», а на іншій — модулюючий безперервний аналоговий сигнал. Частота вихідних імпульсів ШІМ відповідає частоті «зубів» пилки. В ту частину періоду, коли сигнал на позитивному вході вище сигналу на негативному вході, на виході виходить одиниця, в іншу, коли сигнал на позитивному вході нижче сигналу на негативному вході — нуль.

Цифрова ШІМ

У двійковій цифровій техніці, виходи в якій можуть приймати тільки одне з двох значень, наближення бажаного середнього рівня вихідного сигналу за допомогою ШІМ є абсолютно природним. Схема настільки ж проста: пилкоподібний сигнал генерується N-бітовим лічильником.

Фільтрація вихідного сигналу

В тих випадках, де це потрібно, наближення форми вихідного сигналу ШІМ до відповідного йому вхідного сигналу забезпечується інтегруючими елементами, роль яких може відігравати інерційність стану керованого ШІМ елемента (наприклад, печі розжарювання) або для цього використовується інтегруюча ланка — фільтр низьких частот.

Переваги ШІМ

Перемикання відбувається з великою швидкістю, відповідно до типу навантаження, з таким розрахунком, щоб період модульованого сигналу був істотно меншим, ніж інерційність системи, до якої подається сигнал. Частота перемикання може становити декілька разів на хвилину для повільних процесів (наприклад електропечі); 100 Гц для електролампи; від декількох до десятків кГц для електродвигуна або від десятків до сотень кГц для аудіопідсилювача і комп'ютерного блоку живлення.

Для оцінки форми ШІМ сигналу застосовується параметр коефіцієнту заповнення (англ. duty cycle), під яким розуміється відношення тривалості ввімкненого стану (англ. 'on' time) до прийнятого періоду імпульсів; малий коефіцієнт заповнення відповідає режиму енергозберігання, позаяк джерело енергії відключено більшість часу. Коефіцієнт заповнення виражають в відсотках, 100% відповідає ввімкненому стану на весь період.

Головною відзнакою ШІМ є мала втрата енергії на електронному перемикачеві. Він здебільшого перебуває або у вимкненому стані, коли його опір максимальний, напруга максимальна або в режимі насичення — з мінімальним опором, тобто струм максимальний, а падіння напруги на ньому близьке до нуля. ШІМ також органічно вкладається в цифрові технології, велика кількість ШІМ-контролерів виробляється у вигляді мікросхем. Класичним прикладом є мікросхеми UC3842...UC3844.

Цифрова широтно-імпульсна модуляція є різновидом дворівневої ІКМ.

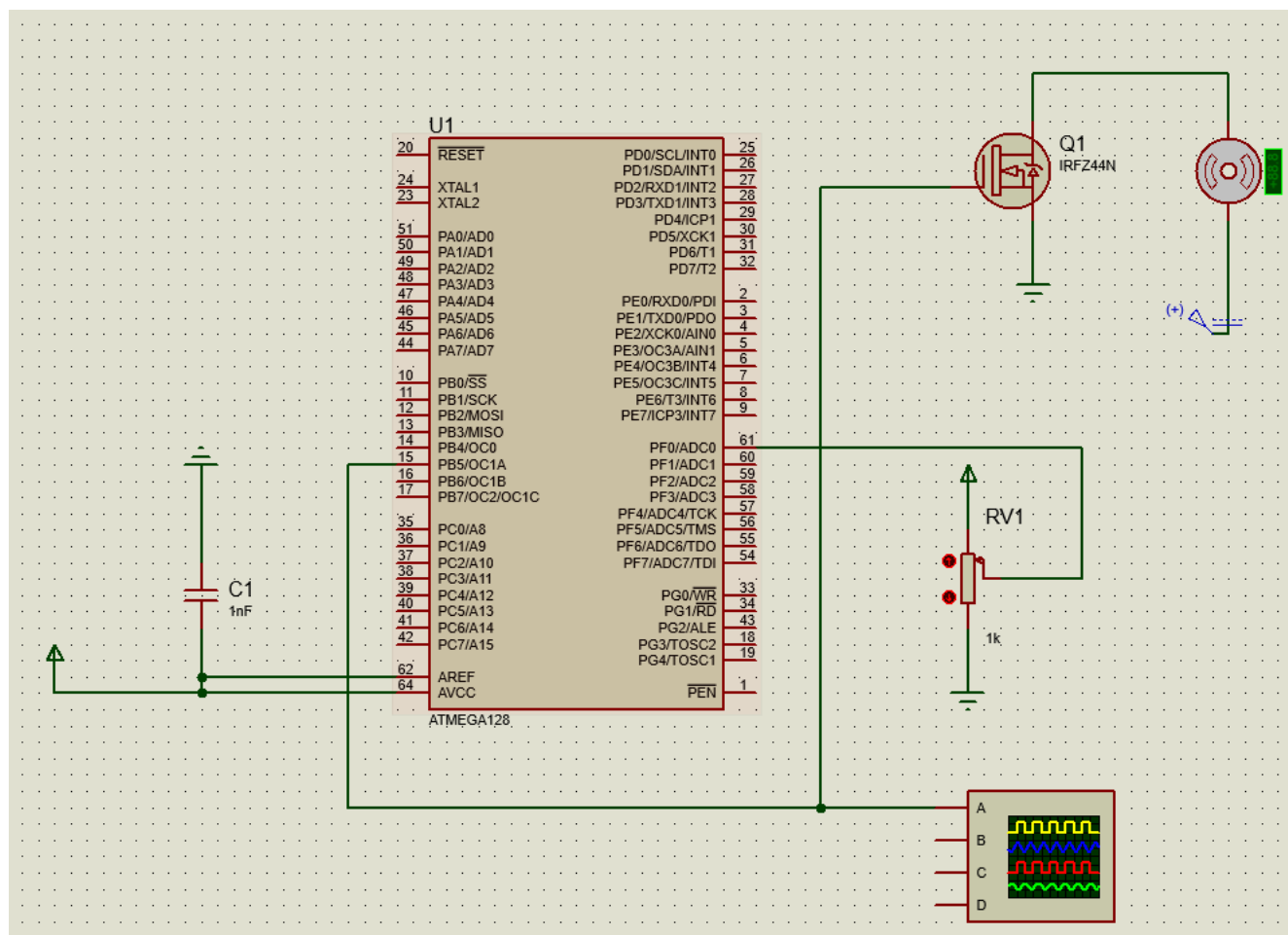


Рис.7.7. Схема керування двигуном за допомогою ШІМ сигналу.

Ця схема використовується для керування двигуном за допомогою ШІМ сигналу на мікроконтролері ATmega128, і включає основні компоненти: мікроконтролер, транзистор

MOSFET для включення і вимикання двигуна, потенціометр для регулювання ширини імпульсу, а також необхідні елементи для стабільної роботи схеми. Ось детальний опис основних компонентів.

7.2 Порядок виконання лабораторної роботи

1. Проаналізувати код в додатку 1
2. Зібрати схему з осцилографом та мікрочіпом Atmega128 та дослідити різні параметри скважності.
3. Зібрати схему керування двигуном рис.7.7.
4. Додати до коду обрахунок скважності з врахуванням значення зі змінного резистору.
5. За допомогою змінного резистора встановити швидкість обертання згідно варіанту.

7.3 Зміст звіту

1. Найменування і мета роботи.
2. Схема з підключеним осцилографом та часова діаграма з нього про данні скважності.
3. Схема симуляції керування двигуном з встановленим значенням швидкості відповідно до варіанту
4. Доповнений код для додатку 2.
5. Висновки по роботі.

7.4 Контрольні запитання

1. Що таке ШІМ сигнал і як він працює?
2. Які основні параметри ШІМ сигналу і що вони означають?
3. Як впливає частота ШІМ сигналу на ефективність роботи пристрою (наприклад, двигуна)?
4. Як за допомогою ШІМ можна ефективно регулювати швидкість обертання двигуна постійного струму?
5. Які типи ШІМ сигналів існують і чим вони відрізняються один від одного (наприклад, неперевернуте та перевернуте)?
6. Які методи використовуються для фільтрації шумів в ШІМ сигналі, особливо при керуванні потужними навантаженнями?

Таблиця 7.1 – Швидкість обертання двигуна

Варіант	1,10	2,11	3,12	4,13	5,14	6,15	7,16	8,17	9,18
Швидкість обертання двигуна об/хв +/-	60	82	103	125	146	163	176	189	210

```
#include <avr/io.h>
#include <util/delay.h>

void pwm_init() {
    // Налаштування таймера 1 в режимі Fast PWM
    TCCR1A = (1 << COM1A1) | (1 << COM1B1) | (1 << COM1C1) | (1 << WGM11); // ШИМ на OC1A,
OC1B, OC1C
    TCCR1B = (1 << WGM13) | (1 << WGM12) | (1 << CS11); // Fast PWM,
прескалер 8
    ICR1 = 19999; // Верхнє значення для таймера (1 кГц частота)
}

void set_duty_cycle(uint16_t duty, uint8_t channel) {
    // Встановлює скважність для кожного каналу
    if (channel == 1) {
        OCR1A = duty; // Скважність на OC1A
    } else if (channel == 2) {
        OCR1B = duty; // Скважність на OC1B
    } else if (channel == 3) {
        OCR1C = duty; // Скважність на OC1C
    }
}

int main() {
    pwm_init();

    // Встановлення виводів як виходи
    DDRB |= (1 << PB5) | (1 << PB6) | (1 << PB7); // Виходи для OC1A, OC1B, OC1C
    DDRC |= (1 << PC0); // Постійний рівень 100% на PC0

    // Скважності

    set_duty_cycle(ICR1 * 3 / 4, 1); // 75% на OC1B
    set_duty_cycle(ICR1 / 2, 2); // 50% на OC1C
    set_duty_cycle(ICR1 / 4, 3); // 25% на OC1B
    PORTC |= (1 << PC0); // Постійний високий рівень (100%)

    while (1) {

    }

    return 0;
}
```

```
#include <avr/io.h>
#include <util/delay.h>

void pwm_init() {
    // Налаштування таймера 1 в режимі Fast PWM
    TCCR1A = (1 << COM1A1) | (1 << WGM11); // ШИМ на OC1A (неінвертуючий режим)
    TCCR1B = (1 << WGM13) | (1 << WGM12) | (1 << CS11); // Fast PWM, прескалер 8
    ICR1 = 19999; // Верхнє значення таймера (для отримання частоти ~1 кГц)
}

void adc_init() {
    // Налаштування ADC
    ADMUX = (1 << REFS0); // Референсне напруження AVCC
    ADCSRA = (1 << ADEN) | (1 << ADPS2) | (1 << ADPS1); // Включення АЦП, прескалер 64
}

uint16_t adc_read(uint8_t channel) {
    // Зчитування значення ADC
    ADMUX = (ADMUX & 0xF0) | (channel & 0x0F); // Вибір каналу
    ADCSRA |= (1 << ADSC); // Запуск конверсії
    while (ADCSRA & (1 << ADSC)); // Очікування завершення
    return ADC;
}

void set_duty_cycle(uint64_t duty) {
    // Встановлення скважності для каналу OC1A
    OCR1A = duty;
}

int main() {
    pwm_init();
    adc_init();

    // Налаштування виводів
    DDRB |= (1 << PB5); // Вихід для PWM (OC1A)

    while (1) {
        uint64_t adc_value = adc_read(0); // Зчитування значення з ADC0 (потенціометр)
        uint64_t duty_cycle = // Додати обрахунок скважності
            set_duty_cycle(duty_cycle); // Встановлення скважності
    }

    return 0;
}
```