

Дослідження роботи ADC0804 та інтегрованого АЦП в мікрочіп Atmega128

Тема: Дослідження особливостей аналого-цифрових перетворень в мікрочіпі Atmega128.

Мета: Дослідити методи та засоби аналого-цифрового перетворення, порівняти різні види АЦП.

Аналого-цифровий перетворювач, АЦП (англ. Analog-to-digital converter, ADC) — пристрій, що перетворює вхідний аналоговий сигнал в дискретний код (цифровий сигнал), який кількісно характеризує амплітуду вхідного сигналу. Зворотне перетворення здійснюється за допомогою цифро-аналогового перетворювача (ЦАП).

Як правило, АЦП — електронний пристрій, що перетворює електричну напругу в двійковий цифровий код.

Розрядність АЦП характеризує кількість дискретних значень, які перетворювач може видати на виході. Вимірюється в бітах. Наприклад, АЦП, здатний видати 256 дискретних значень (0..255), має розрядність 8 бітів.

Розрядність може бути також визначена в величинах вхідного сигналу і виражена, наприклад, у вольтах. Розрядність за напругою дорівнює напрузі, що відповідає максимальному вихідному коду, який ділиться на кількість вихідних дискретних значень. Наприклад:

- Приклад 1
 - Діапазон вхідних значень = від 0 до 10 вольт
 - Розрядність АЦП 12 бітів: $2^{12} = 4096$ рівнів квантування
 - Розрядність за напругою: $(10-0)/4096 = 0.00244$ вольт = 2.44 мВ

- Приклад 2
 - Діапазон вхідних значень = від -10 до +10 вольт
 - Розрядність АЦП 14 бітів: $2^{14} = 16384$ рівнів квантування
 - Розрядність за напругою: $(10-(-10))/16384 = 20/16384 = 0.00122$ вольт = 1.22 мВ

На практиці розрядність АЦП обмежена співвідношенням сигнал/шум вхідного сигналу. При великій інтенсивності шумів на вході АЦП розрізнення сусідніх рівнів вхідного сигналу стає неможливим, тобто погіршується розрядність. При цьому реальний досяжний дозвіл описується ефективною кількістю розрядів (effective number of bits — ENOB), яка менше, ніж реальна розрядність АЦП. При перетворенні сильно зашумленого сигналу молодші біти вихідного коду практично непридатні, оскільки містять шум. Для досягнення заявленої розрядності співвідношення С/Ш вхідного сигналу повинне бути приблизно 6 дБ на кожен біт розрядності.

Більшість АЦП вважаються лінійними, хоча аналого-цифрове перетворення по суті є нелінійним процесом (оскільки операція перетворення безперервного сигналу в дискретний — операція необоротна і, отже, нелінійна). Термін лінійний стосовно АЦП означає, що діапазон вхідних значень, що відображається на вихідне цифрове значення, зв'язаний за лінійним законом з цим вихідним значенням, тобто вихідне значення k досягається при діапазоні вхідних значень від $m(k + b)$ до $m(k + 1 + b)$, де m і b — деякі константи. Константа b , як правило, має значення 0 або -0.5. Якщо $b = 0$, АЦП називають mid-rise, якщо ж $b = -0.5$, то АЦП називають mid-tread.

Якби густина ймовірності амплітуди вхідного сигналу мала рівномірний розподіл, то співвідношення сигнал/шум (стосовно шуму квантування) було б максимально можливим.

З цієї причини зазвичай перед квантуванням за амплітудою сигнал пропускають через безінерційний перетворювач, передавальна функція якого повторює функцію розподілу самого сигналу. Це покращує достовірність передачі сигналу, оскільки найважливіші області амплітуди сигналу квантуються з кращою розрядністю. Відповідно, при цифро-аналоговому перетворенні потрібно буде обробити сигнал функцією, зворотною функції розподілу початкового сигналу.

Цей принцип також використовується в компандерах, які застосовуються у магнітофонах і різних комунікаційних системах з метою максимізації ентропії.

Наприклад, голосовий сигнал має лапласовий розподіл амплітуди. Це означає, що сигнал в області малих амплітуд несе більше інформації, ніж в області великих амплітуд. З цієї причини логарифмічні АЦП часто застосовуються в системах передачі голосу для збільшення динамічного діапазону значень, що передаються без зміни якості передачі сигналу в області малих амплітуд.

8-бітові логарифмічні АЦП з а-законом або μ -законом забезпечують широкий динамічний діапазон і мають високий дозвіл в найкритичнішому діапазоні малих амплітуд; лінійний АЦП з подібною якістю передачі повинен був би мати розрядність близько 12 біт.

Є декілька джерел похибки АЦП. Похибки квантування і (вважаючи, що АЦП повинен бути лінійним) нелінійності властиві будь-якому аналого-цифровому перетворенню. Крім того, існують так звані апертурні помилки які є наслідком джитера (англ. jitter) тактового генератора, вони виявляються при перетворенні сигналу в цілому (а не одного відліку).

Ці похибки вимірюються в одиницях, званих МЗР — молодший значущий розряд. У приведену вище прикладі 8-бітового АЦП помилка в 1 МЗР становить 1/256 від повного діапазону сигналу, тобто 0.4 %.

Похибка повної шкали - відносна різниця між реальним і ідеальним значеннями межі шкали перетворення (FSO, англ. Full-Span Output) при відсутності зміщення нуля. Ця похибка є мультиплікативною складовою повної похибки. Іноді вказується відповідним числом МЗР.

Похибка зсуву нуля - значення $U_{вх}$, коли вхідний код ЦАП дорівнює нулю. Є адитивною складовою повної похибки.

Похибки квантування є наслідком обмеженої розрядності АЦП. Цей недолік не може бути усунено за жодного типу аналого-цифрового перетворення. Абсолютна величина помилки квантування в кожного відліку знаходиться в межах від нуля до половини МЗР.

Як правило, амплітуда вхідного сигналу значно більша, ніж МЗР. В цьому випадку помилка квантування не корельована з сигналом і має рівномірний розподіл. Її середньоквадратичне значення збігається з середньоквадратичним

відхиленням розподілу, який дорівнює $\frac{1}{\sqrt{12}} \text{LSB} \approx 0.289 \text{LSB}$.

У випадку 8-бітового АЦП це складе 0.113 % від повного діапазону сигналу.

Усім АЦП властиві помилки, пов'язані з нелінійністю, які є наслідком фізичної недосконалості АЦП. Це призводить до того, що передавальна характеристика (у вказаному вище сенсі) відрізняється від лінійної (точніше від бажаної функції, оскільки вона не обов'язково лінійна). Помилки можуть бути зменшені шляхом калібрування.

Важливим параметром, що описує нелінійність, є інтегральна нелінійність (INL) і диференціальна нелінійність (DNL). В ідеальному випадку відліки беруться через рівні проміжки часу. Проте, в реальності час моменту узяття відліку схильний до флуктуацій

внаслідок тремтіння фронту синхросигналу (clock jitter). Вважаючи, що невизначеність моменту часу взяття відліку порядку, отримуємо, що помилка, обумовлена цим явищем, може бути оцінена як

$$E_{ap} \leq |x'(t)\Delta t| \leq 2A\pi f_0 \Delta t$$

Легко бачити, що помилка відносно невелика на низьких частотах, проте на великих частотах вона може істотно зрости. Ефект апертурної похибки може бути проігнорований, якщо її величина порівняно невелика в порівнянні з помилкою квантування. Таким чином, можна встановити такі вимоги до тремтіння фронту сигналу синхронізації:

$$\Delta t < \frac{1}{2^q \pi f_0}$$

де q — розрядність АЦП.

Аналоговий сигнал є неперервною функцією часу, в АЦП він перетвориться в послідовність цифрових значень. Отже, необхідно визначити частоту вибірки цифрових значень з аналогового сигналу. Частота, з якою проводяться вибірки, отримала назву частота дискретизації (Sampling rate) АЦП.

Безперервно змінний сигнал з обмеженою спектральною смугою піддається оцифруванню (тобто значення сигналу вимірюються через інтервал часу T — період дискретизації) і початковий сигнал може бути точно відновлений з дискретних в часі значень шляхом інтерполяції. Точність відновлення обмежена помилкою квантування. Проте, відповідно до теореми Котельникова-Шеннона точне відновлення можливе тільки якщо частота дискретизації вища, ніж подвоєна максимальна частота в спектрі сигналу.

Оскільки реальні АЦП не можуть провести аналого-цифрове перетворення миттєво, вхідне аналогове значення повинне утримуватися постійним принаймні від початку до кінця процесу перетворення (цей інтервал часу називають час перетворення). Це завдання вирішується шляхом використання спеціальної схеми на вході АЦП — пристрій вибірки-зберігання (ПВЗ, рос. УВХ). ПВЗ, як правило, зберігає вхідну напругу в конденсаторі, сполученому з входом через аналоговий ключ: при замиканні ключа відбувається вибірка вхідного сигналу (конденсатор заряджає до вхідної напруги), при розмиканні — зберігання. Багато АЦП, виконані у вигляді інтегральних мікросхем містять вбудовані ПВЗ.

ADC0804 — це 8-бітний аналогово-цифровий перетворювач (АЦП), що використовується для перетворення аналогового сигналу у цифровий. Він підтримує діапазон вхідних сигналів від 0 до 5 В і має час перетворення близько 100 мікросекунд, що робить його підходящим для багатьох додатків із помірною швидкістю перетворення.

Основні характеристики ADC0804:

Розрядність: 8 біт.

Діапазон вхідної напруги: від 0 до 5 В.

Максимальний час перетворення: 100 мікросекунд.

Тип інтерфейсу: паралельний, що дозволяє передавати дані 8-бітними словами.

Споживана потужність: низька, підходить для низьковольтних і енергозберігаючих систем.

Операційна напруга: 5 В (логічний рівень також становить 5 В).

ADC0804 широко використовується в електронних системах, де потрібно вимірювати фізичні величини, такі як температура, тиск, вологість тощо. Ці величини зазвичай представляються у вигляді аналогових сигналів, які цей АЦП перетворює в цифровий формат для подальшої обробки мікроконтролером або мікропроцесором. Типові області застосування: системи автоматизації, промислові датчики та сенсори, пристрої контролю та моніторингу, аудіообладнання, цифрові вимірювальні прилади.

Розпіновка ADC0804:

Vcc (Pin 20): Живлення АЦП (5 В).

GND (Pin 10): Заземлення.

Vin+ (Pin 6): Позитивний вхідний аналоговий сигнал, що підлягає оцифруванню (аналоговий сигнал від датчика).

Vin- (Pin 7): Негативний вхідний сигнал. Зазвичай з'єднується із заземленням, коли використовується одинарний сигнал.

CS (Chip Select, Pin 1): Активно-низький вхід. Якщо цей пін низький, АЦП активний.

RD (Read, Pin 2): Активно-низький вхід для зчитування даних. Під час низького стану АЦП видає дані на шину даних.

WR (Write, Pin 3): Активно-низький вхід для ініціації процесу перетворення аналогового сигналу в цифровий.

INTR (Interrupt, Pin 5): Активно-низький вихід, який сигналізує про завершення перетворення. Після перетворення цей пін стає низьким.

CLK IN (Pin 4): Вхід тактового сигналу. Якщо тактовий генератор не використовується, то можна подати зовнішній генератор на цей пін.

D0-D7 (Pins 11-18): 8-бітна шина даних. Використовується для передачі результатів перетворення.

Vref/2 (Pin 9): Напруга опорного рівня. Встановлює максимальне значення вхідного сигналу (по замовчуванню це 5 В, але може бути змінене).

Підключення ADC0804:

Живлення: Підключаємо Vcc до +5 В і GND до заземлення.

Вхідний сигнал: Подаємо аналоговий сигнал на Vin+ (Pin 6). Якщо використовується одинарний сигнал, Vin- підключається до GND.

Тактовий сигнал: Якщо використовується зовнішній тактовий сигнал, його подаємо на пін CLK IN (Pin 4).

Ініціація перетворення: Щоб почати процес перетворення, пін WR (Pin 3) переводиться у низький стан. Після завершення перетворення INTR (Pin 5) стане низьким.

Зчитування результату: Після завершення перетворення INTR сигналізує про готовність даних, і пін RD (Pin 2) переводиться у низький стан для зчитування результатів із D0-D7.

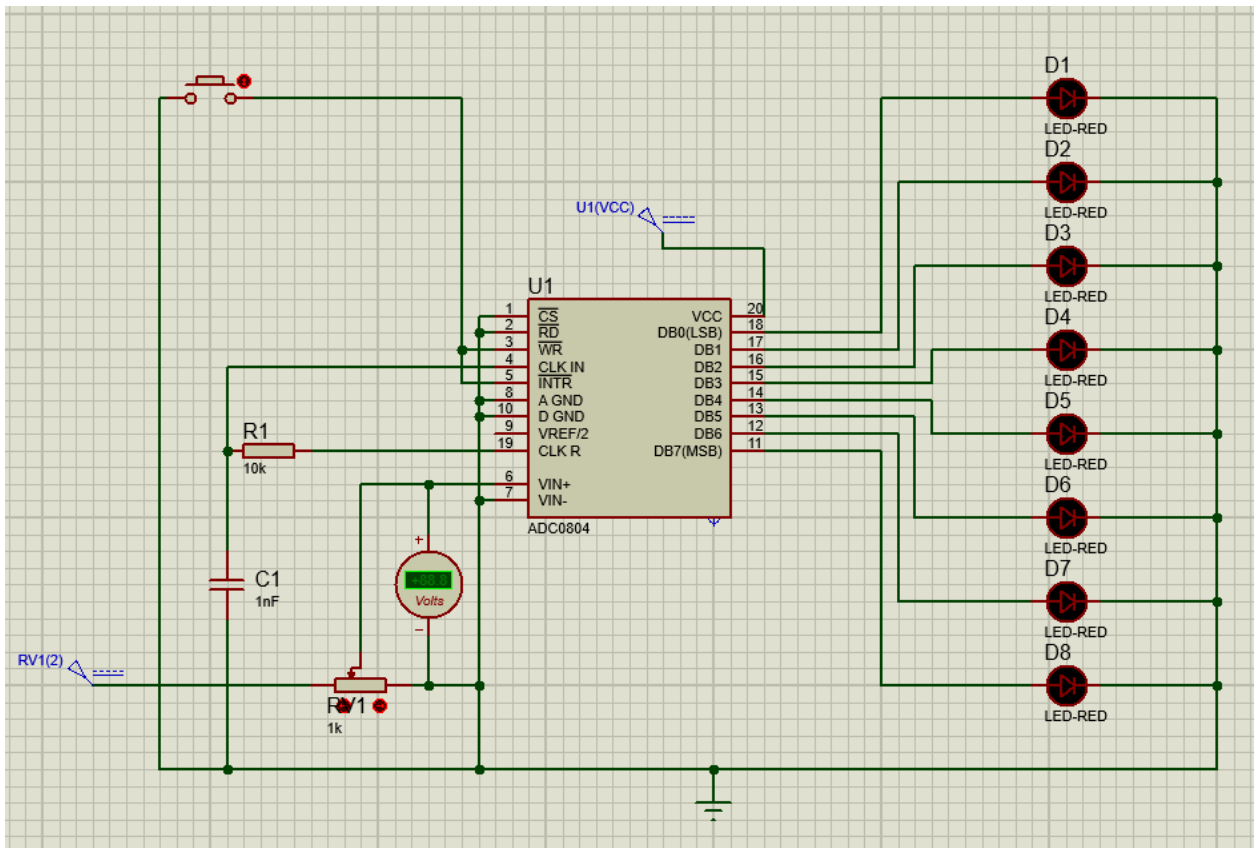


Рис.3.1 – Підключення ADC0804.

За даним підключенням індикатори покажуть значення де D8 – старші біти, D1 – молодші біти (3.1).

$$D = \frac{V_{IN+}}{V_{REF}} \times 255, \quad (3.1)$$

де - D — це цифрове значення з АЦП, V_{IN+} — вхідна аналогова напруга, V_{REF} — опорна напруга. Отримане двійкове значення потрібно перевести в десяткове число, після чого поділити на кількість можливих значень в даному АЦП та помножити на вхідну напругу. Так отримаємо значення вимірювальної величини(3.2).

$$V_{IN+} = \frac{D}{255} \times V_{REF}, \quad (3.2)$$

де: V_{IN+} — напруга, що відповідає цифровому значенню D, D — цифрове значення, V_{REF} - опорна напруга.

В мікроконтролері ATmega128 вбудований 10-бітний аналогово-цифровий перетворювач (АЦП), який може використовуватись для перетворення аналогових сигналів у цифрові значення для подальшої обробки.

Основні характеристики АЦП в ATmega128:

Розрядність: 10 біт (це означає, що діапазон вихідних цифрових значень становить від 0 до 1023).

Кількість каналів: 8 мультиплексованих каналів, що дозволяє підключати до 8 різних аналогових входів.

Діапазон вхідної напруги: Вхідний сигнал може бути в діапазоні від 0 до напруги живлення (V_{CC}), зазвичай це 5 В або 3,3 В.

Опорна напруга: Можна вибрати зовнішню або внутрішню опорну напругу. Є внутрішня опорна напруга 2,56 В.

Швидкість перетворення: Швидкість залежить від тактової частоти, яка подається на АЦП, але зазвичай становить близько 13-260 мікросекунд на перетворення.

Режими роботи: Підтримується як одиничне перетворення, так і безперервний режим перетворення.

Можливість використання режиму з шумозаглушенням: Мікроконтролер ATmega128 може перемикатися в режим зниженого шуму під час роботи АЦП для підвищення точності.

Розпіновка каналів АЦП: ADC0-ADC7: Це вхідні пін-контакти для аналогових сигналів. Вони відповідають пін-контактам від PF0 до PF7 на мікроконтролері.

AREF: Це пін для підключення зовнішньої опорної напруги.

AVCC: Напруга живлення для АЦП, зазвичай підключена до того ж джерела, що й V_{CC} мікроконтролера.

Програмне налаштування АЦП: АЦП налаштовується і керується через кілька спеціальних регістрів. Нижче описані ключові регістри: ADMUX (ADC Multiplexer Selection Register):

Використовується для вибору вхідного каналу (ADC0-ADC7) та налаштування опорної напруги.

Фільтрація живлення: Для стабільної роботи АЦП рекомендується додати конденсатор між AVCC та GND, щоб уникнути шумів на лінії живлення.

Опорна напруга: Якщо важлива висока точність, варто використовувати стабільне джерело опорної напруги.

ATmega128 надає простий і зручний спосіб перетворювати аналогові сигнали в цифрові для багатьох видів сенсорних застосувань, де важлива інтеграція з аналоговими датчиками.

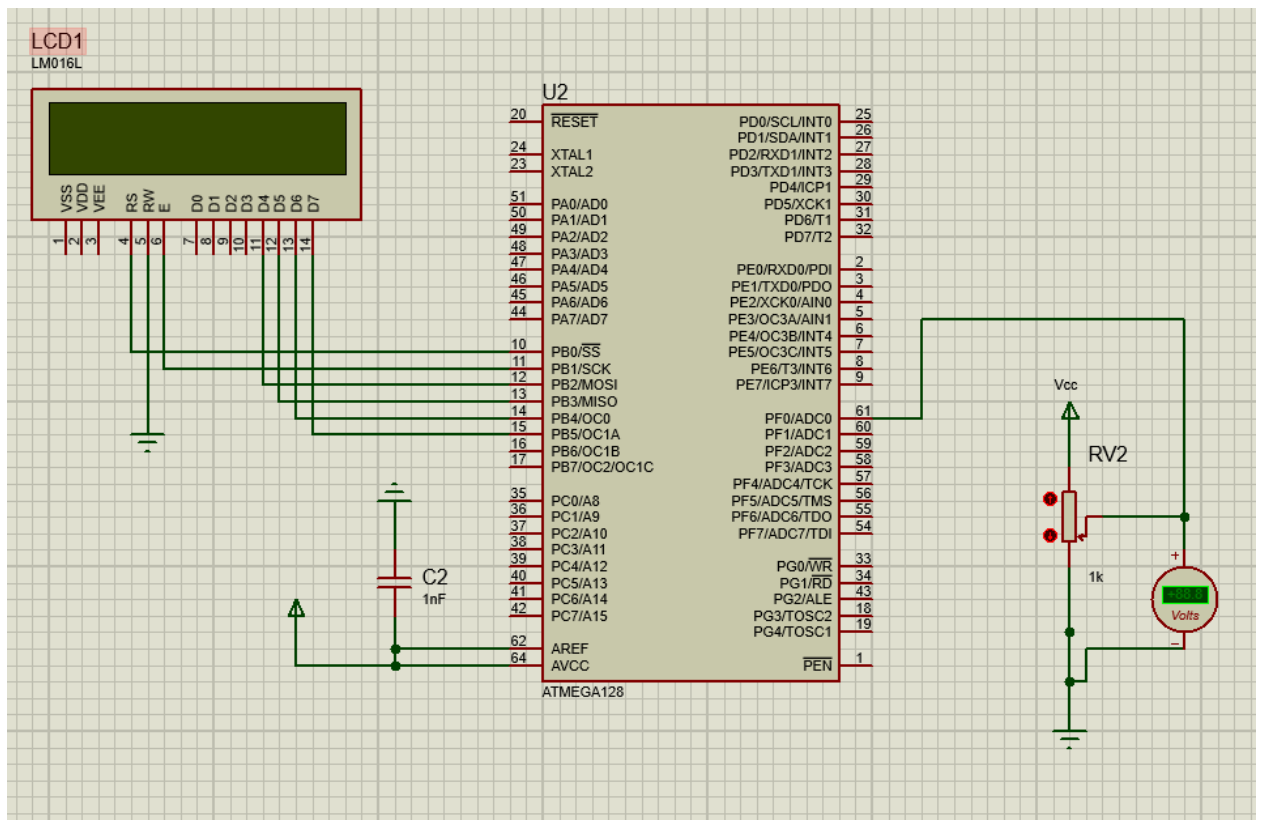


Рис.3.2 – Активація вбудованого АЦП в мікросхемі Atmega128.

Обрахунок напруги здійснюється відповідно до формул (3.1, 3.2) з врахуванням кількості рівнів квантування АЦП.

3.2 Порядок виконання лабораторної роботи

1. Ознайомитись з теоретичними відомостями.
2. Побудувати схему зображену на рисунку 3.1.
3. Задати значення вхідної напруги відповідно до варіанту та обрахувати вихідне значення АЦП.
4. Повторити пункт 2 та 3 для вбудованого АЦП в мікросхемі Atmega128, відобразити вимірне значення в другому рядку, а в першому своє прізвище та ім'я. Використати додаток 1.
5. Об'єднати в одну схему АЦП ADC0804 та Atmega128, виміряти одне й те саме значення напруги та порівняти вихідні дані.
6. Зробити висновки .

3.3 Зміст звіту

1. Найменування і мета роботи.
2. Схема підключення та обрахунок вихідного значення напруги для ADC0804.
3. Схема підключення та код для Atmega128.
4. Схема підключення Atmega128, ADC0804 та обчислені дані.
5. Висновки по роботі.

3.4 Контрольні запитання

1. Яку розрядність має вбудований АЦП у мікроконтролері ATmega128?
2. Скільки аналогових каналів доступно для використання з АЦП у ATmega128?
3. Який діапазон вхідної напруги підтримує АЦП у ATmega128?
4. Які налаштування треба внести, щоб вибрати зовнішню опорну напругу для АЦП?
5. Який біт у регістрі ADCSRA відповідає за ввімкнення АЦП?
6. Чому важливо використовувати фільтрацію живлення для стабільної роботи АЦП?


```

#define F_CPU 1000000UL // Частота мікроконтролера 1 МГц
#include <avr/io.h>
#include <util/delay.h>
// Піни для підключення дисплея
#define RS PB0
#define EN PB1
#define D4 PB2
#define D5 PB3
#define D6 PB4
#define D7 PB5
// Функції для керування дисплеєм
void LCD_Command(unsigned char cmd);
void LCD_Char(unsigned char data);
void LCD_Init(void);
void LCD_String(const char *str);
void LCD_Clear(void);
// Встановлення порту і піна для дисплея
void LCD_Port(unsigned char data) {
    if (data & 1) PORTB |= (1<<D4);
    else PORTB &= ~(1<<D4);
    if (data & 2) PORTB |= (1<<D5);
    else PORTB &= ~(1<<D5);
    if (data & 4) PORTB |= (1<<D6);
    else PORTB &= ~(1<<D6);
    if (data & 8) PORTB |= (1<<D7);
    else PORTB &= ~(1<<D7);
}
void LCD_Command(unsigned char cmd) {
    // Відправляємо команду (RS=0)
    PORTB &= ~(1<<RS);
    // Передаємо старші 4 біти
    LCD_Port(cmd >> 4);
    PORTB |= (1<<EN);
    _delay_ms(1);
    PORTB &= ~(1<<EN);
    // Передаємо молодші 4 біти
    LCD_Port(cmd);
    PORTB |= (1<<EN);
    _delay_ms(1);
    PORTB &= ~(1<<EN);
}
void LCD_Char(unsigned char data) {
    // Відправляємо дані (RS=1)
    PORTB |= (1<<RS);
    // Передаємо старші 4 біти
    LCD_Port(data >> 4);
    PORTB |= (1<<EN);
    _delay_ms(1);
    PORTB &= ~(1<<EN);
    // Передаємо молодші 4 біти
    LCD_Port(data);
    PORTB |= (1<<EN);
    _delay_ms(1);
    PORTB &= ~(1<<EN);
}
void LCD_Init(void) {
    // Налаштовуємо пін як вихід
    DDRB |= (1<<RS) | (1<<EN) | (1<<D4) | (1<<D5) | (1<<D6) | (1<<D7);
    // Початкова ініціалізація
    LCD_Command(0x02); // Перемикає в 4-бітний режим
    LCD_Command(0x28); // 2 рядки, 5x8 точок
    LCD_Command(0x0C); // Увімкнути дисплей, без курсора
    LCD_Command(0x06); // Інкрементувати курсор
}

```

```

        LCD_Command(0x01); // Очистити дисплей
        _delay_ms(2);
    }
    void LCD_String(const char *str) {
        while (*str) {
            LCD_Char(*str++);
        }
    }
    void LCD_Clear(void) {
        LCD_Command(0x01); // Команда очищення дисплея
        _delay_ms(2);
    }

    // Ініціалізація АЦП
    void adc_init() {
        ADMUX = (1<<REFS0); // Використовуємо внутрішнє опорне напруження AVCC
        ADCSRA = (1<<ADEN) | (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0); // Вмикаємо АЦП та
        встановлюємо преддільник на 128
    }

    // Читання значення з АЦП
    uint16_t adc_read(uint8_t channel) {
        channel &= 0x07; // Обмежуємо номер каналу до діапазону [0..7]
        ADMUX = (ADMUX & 0xF8) | channel; // Вибираємо канал (ADC0 на PF0)
        ADCSRA |= (1<<ADSC); // Починаємо перетворення
        while (ADCSRA & (1<<ADSC)); // Чекаємо завершення перетворення
        return ADC; // Повертаємо результат
    }

    int main(void) {
        // Ініціалізація дисплея
        LCD_Init();
        adc_init();

        while (1) {
            uint16_t adc_value = adc_read(0); // Читаємо з PF0 (ADC0)

            // Очищення дисплея перед виведенням
            LCD_Clear();

            char buffer[10];
            itoa(adc_value, buffer, 10); // Конвертуємо значення в рядок

            // Виводимо дані на дисплей
            LCD_String("ADC Value: ");
            LCD_String(buffer);

            _delay_ms(500); // Затримка на 500 мс
        }
    }
}

```

Варіант	1,11,21	2,12,22	3,13,23	4,14,24	5,15,25	6,16,26	7,17,27	8,18,28	9,19,29	10,20,30
Значення Вхідної наруги	0.5V	1V	1.5V	2V	2.5V	3V	3.5V	4V	4.5V	5V