

Курс Arduino для початківців

ПЛАТФОРМА ARDUINO ТА ВИВЕДЕННЯ ІНФОРМАЦІЇ

План заняття

- ▶ Вирішення поточних питань та орг. моментів
 - ▶ Опрацювання домашнього завдання
 - ▶ Використання коментарів в коді
-
- ▶ Пристрої виведення
 - ▶ Цифрове та аналогове виведення
 - ▶ Широтно Імпульсна Модуляція
 - ▶ Основні функції для роботи з цифровими портами
 - ▶ Робота та конфігурація портів для виведення інформації
 - ▶ Написання програми "Wave_Arduino"
 - ▶ Написання програми "PWM_Arduino"

Використання коментарів в коді

Коментарі - це рядки в програмі, які використовуються для інформування вас самих або інших про те, як працює програма. Вони ігноруються компілятором і не експортуються в процесор, таким чином, вони не займають місце в пам'яті мікроконтролера Atmega.

► Підказка

Під час експериментів з кодом, «закоментування» частин програми - вірний спосіб видалення рядків, в яких можуть бути помилки. Так рядки в коді залишаються, але перетворюються в коментарі, і компілятор просто ігнорує їх. Це може бути особливо корисно при локалізації проблеми, або коли не виходить скомпілювати програму, а повідомлення про помилку при компіляції приховано.

Приклад використання коментарів в коді

- ▶ Коментарі призначені тільки для того, щоб допомогти вам зрозуміти (або згадати), як працює ваша програма або пояснити це іншим. Є два способи помітити рядок як коментар:

`x = 5; // Це коментар в одному рядку. Все після подвійного слеша - коментар`

`/* Це багаторядковий коментар - використовуйте його для закоментірованія цілих шматків коду`

`if (gwb == 0) {`

`// коментар в рядку допустимо всередині багаторядкового комента`

`// Але не інший багатостроковий коментар`

`}`

`* /`

Пристрої виведення

Пристрої виведення - периферійні пристрої, що перетворюють результати обробки цифрових машинних кодів в форму, зручну для сприйняття людиною або придатну для впливу на виконавчі органи об'єкта управління.

- ▶ Пристрої для виведення візуальної інформації (монітор, дисплей, принтер, плоттер, проектор)



- ▶ Пристрої для виведення звукової інформації (вбудований динамік, колонки, навушники)



- ▶ Пристрої введення-виведення (дисковод, жорсткий диск, різні порти, мережеві інтерфейси)

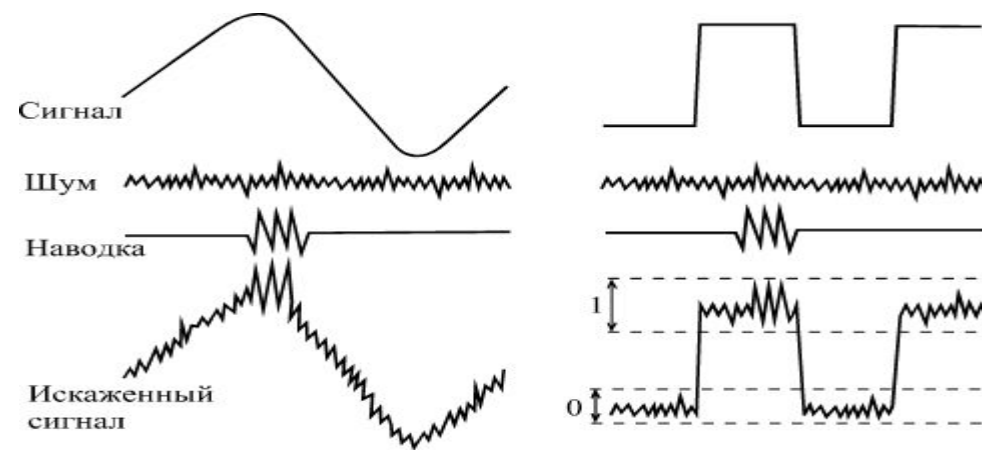
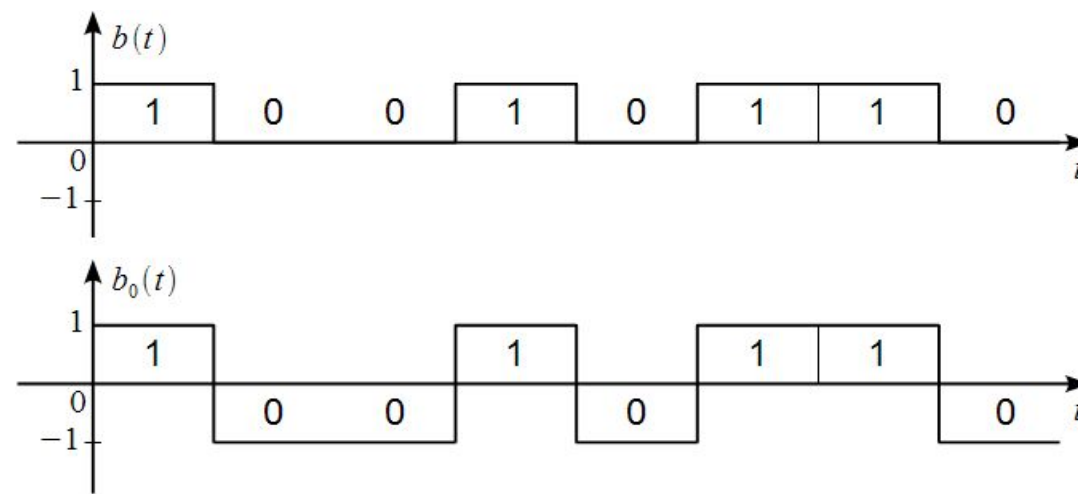


Цифровий та аналоговий сигнал

Цифровий сигнал - сигнал, який можна представити у вигляді послідовності дискретних (цифрових) значень. У наш час найбільш поширені виконавчі цифрові сигнали в зв'язку з простотою кодування і використовуваних в двійковій електроніці.

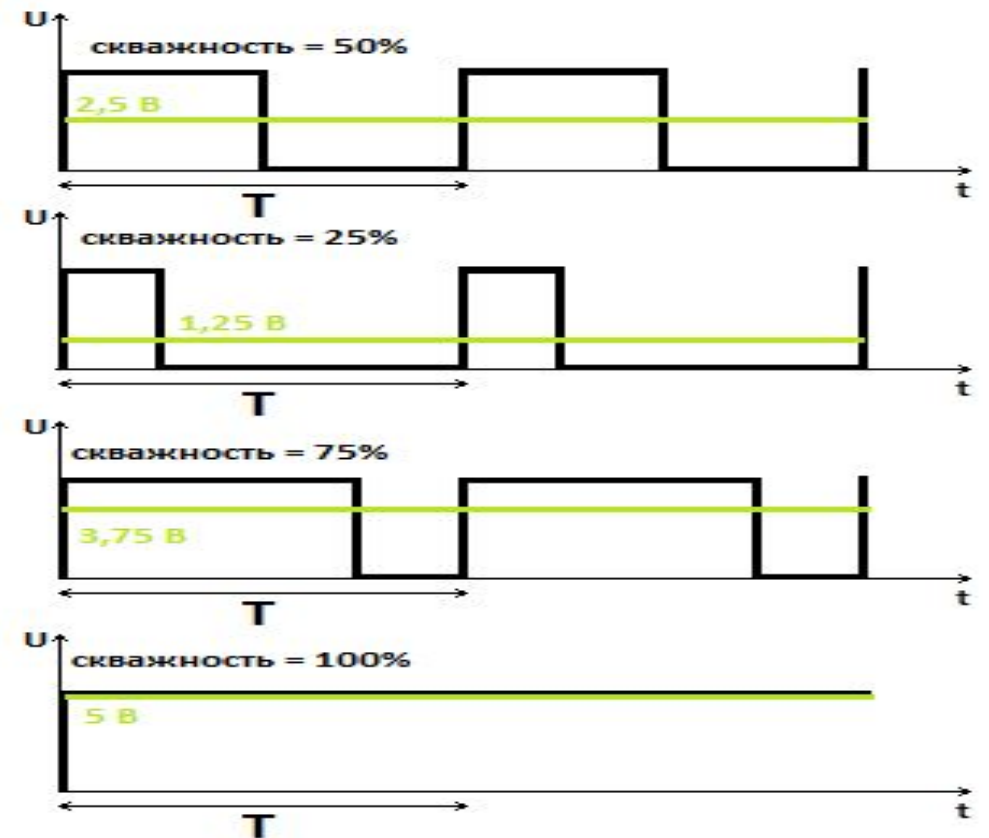
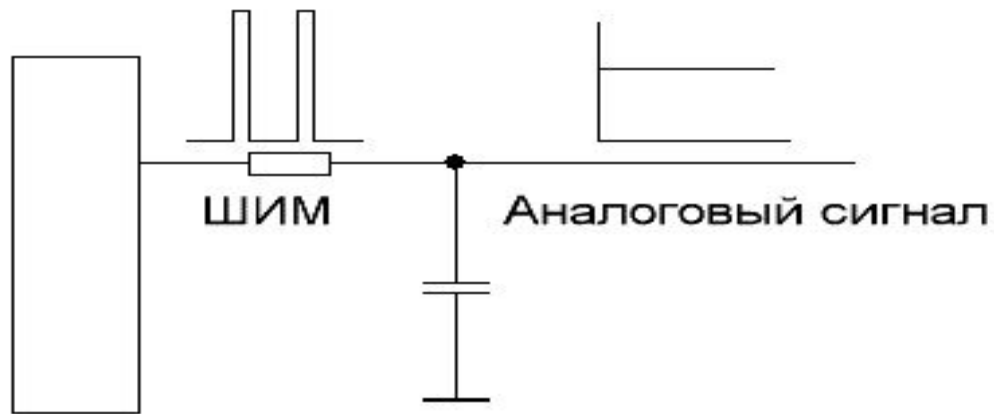
Для передачі цифрового сигналу по аналоговим каналам (наприклад, електричним або радіоканалах) використовуються різні види модуляції.

Аналоговий сигнал - сигнал даних, у якого кожен з представлених параметрів описується функцією часу і безперервною множиною можливих значень.



Широтно Імпульсна Модуляція

Широтно-імпульсна модуляція - процес управління потужністю, що підводиться до навантаження, шляхом зміни скважності імпульсів, при постійній частоті. Розрізняють аналогову ШІМ і цифрову ШІМ.



Основні функції для роботи з портами `analogWrite` та `digitalWrite`

- ▶ `analogWrite()`

Формує заданий аналогове напруга на виводі у вигляді ШІМ-сигналу. Після виклику `analogWrite()`, на виводі буде безперервно генеруватися ШІМ-сигнал з заданим коефіцієнтом заповнення до наступного виклику функції `analogWrite()` (або до моменту виклику `digitalRead()` або `digitalWrite()`, взаємодіючих з цим же висновком). Частота ШІМ становить приблизно 490 Гц.

- ▶ Функція `analogWrite()` не має нічого спільного з аналоговими виходами і функцією `analogRead()`.
- ▶ При роботі з `analogWrite()` попередній виклик функції `pinMode()` для перемикання виводів в режим «вихід» не потрібно.

Синтаксис функції analogWrite

- ▶ `analogWrite (pin, value)`

Параметри:

- ▶ `pin`: порт, на якому буде формуватися ШІМ.
- ▶ `value`: коефіцієнт заповнення - лежить в межах від 0 (завжди вимкнений) до 255 (завжди включений).

На виході 5 і 6 генерується ШІМ-сигнал з коефіцієнтом заповнення більшим, ніж задане очікуване значення. Це відбувається в результаті взаємодії з функціями `millis ()` і `delay ()`, які використовують той же внутрішній таймер, що застосовується для генерування ШІМ-сигналу.

Функція для роботи з цифровими портами digitalWrite

- ▶ Відправляє на цифровий порт значення HIGH або LOW.
- ▶ Якщо функцією pinMode () порт налаштований як (OUTPUT), то при виконанні функції digitalWrite () його напруга буде змінена на відповідне значення: 5 В (або 3.3 В для плат, що працюють від 3.3 В) при відправці HIGH, 0 В (земля) - при LOW.
- ▶ Якщо порт налаштований як вхід INPUT, то відправка функцією digitalWrite () значення HIGH призведе до підключення внутрішнього підтягуючого резистора номіналом 20 кОм. Запис значення LOW призведе до відключення підтяжки. Внутрішній підтягуючий резистор може забезпечити тільки тьмяне світіння світлодіода. Тому, якщо світлодіод горить, але дуже тьмяно, найбільш ймовірна причина цього - підтягаючий резистор. Для вирішення даної проблеми необхідно перевести відповідний порт в режим виходу за допомогою функції pinMode () .

Синтаксис функції digitalWrite

- ▶ digitalWrite (pin, value)

Параметри:

- ▶ pin: номер виводу
- ▶ value: значення HIGH або LOW

ПРИМІТКА: Існують деякі складності при використанні порта 13 в якості цифрового входу. Причиною цього є світлодіод і резистор, які припаяні до цього порта на більшості плат Ардуіно. При включенні внутрішнього підтягуючого резистора 20 кОм, напруга на цьому встановиться на рівні близько 1.7 В, замість очікуваних 5 В, оскільки світлодіод і послідовно з'єднаний резистор на платі знижують рівень напруги. Таким чином, порт буде завжди знаходитися в стані LOW. Тому, щоб використовувати вихід 13 в якості цифрового входу, необхідно використовувати зовнішній резистор на землю.

Написання програми “Wave_Arduino”

```
int ledPin = 13;           // світодиод підключен к виводу 13

void setup()
{
    pinMode(ledPin, OUTPUT); // переключаем цифровой вывод в режим выхода
}

void loop()
{
    digitalWrite(ledPin, HIGH); // включаем светодиод
    delay(1000);                // ждем 1 секунду
    digitalWrite(ledPin, LOW);  // выключаем светодиод
    delay(1000);                // ждем 1 секунду
}
```

Написання програми “PWM_Arduino”

```
int ledPin = 9;    // светодиод подключен к цифровому выводу 9
int analogPin = 3; // потенциометр подключен к аналоговому выводу 3
int val = 0;       // переменная для хранения считанного значения

void setup()
{
    pinMode(ledPin, OUTPUT); // переключение вывода в режим «выход»
}

void loop()
{
    val = analogRead(analogPin); // считываем входное напряжение
    analogWrite(ledPin, val / 4); /* значения, возвращаемые analogRead лежат в пределах от 0 до 1023,
                                   а задаваемый коэффициент analogWrite – от 0 to 255*/
}
```