

Практичне заняття №5.

Програмування генератора вимірювальних сигналів з використанням Arduino

Мета: дослідити можливості створення генератора вимірювальних сигналів, за допомогою спеціальних бібліотек, в Arduino.

Завдання: зібрати схему і написати програму створення генератора вимірювальних сигналів, а також відтворення тональних сигналів.

Обладнання: плата Arduino; маленький звуковий динамік із опором 8 Ом; резистор 100 Ом; з'єднувальні проводи; ультразвуковий дальномір; кнопки.

Загальні відомості

Для відтворення звукового сигналу, використовуючи мікроконтролер Arduino, використовують команду *tone()*.

tone() генерує на виводі мікроконтролера прямокутний сигнал заданої частоти (з коефіцієнтом заповнення 50%). Функція також дозволяє задавати тривалість сигналу. Однак, якщо тривалість сигналу не вказана, він буде генеруватися доти, поки не буде викликана функція *noTone()*. Для відтворення звуку вивід можна підключити до зумеру або динаміку.

У кожен момент часу може генеруватися тільки один сигнал заданої частоти. Якщо сигнал вже генерується на будь-якому виводі, то використання функції *tone()* для цього виводу просто призведе до зміни частоти цього сигналу. У той же час виклик функції *tone()* для іншого виводу не матиме ніякого ефекту.

Використання функції *tone()* може впливати на ШІМ-сигнал на виводах 3 і 11 (на всіх платах, крім Mega).

Для відтворення різних звуків на кількох виводах, необхідно спершу викликати *noTone()* на одному і тільки після цього використовувати функцію *tone()* на наступному.

Синтаксис:

tone (pin, frequency),

tone (pin, frequency, duration).

Параметри:

pin: вивід, на якому буде генеруватися сигнал;

frequency: частота сигналу в Герцах – unsigned int;

duration: тривалість сигналу в мілісекундах (опціонально) – unsigned long.

Значення, що повертає – немає.

Приклад коду програми формування тонального сигналу

Наведений нижче код використовує додатковий файл, pitches.h. Цей файл містить всі значення кроку для простих тональних сигналів. Наприклад, NOTE_C4 є нотою C₄. NOTE_FS 4 – фа-дієз, і так далі.

Отож, спочатку треба створити файл pitches.h, у котрому зазначена нота та відповідна їй частота, й зберегти його у папці з проектом. Файл має в собі містити наступний код:

```
#define NOTE_B0 31
#define NOTE_C1 33
#define NOTE_CS1 35
#define NOTE_D1 37
#define NOTE_DS1 39
#define NOTE_E1 41
#define NOTE_F1 44
#define NOTE_FS1 46
#define NOTE_G1 49
#define NOTE_GS1 52
#define NOTE_A1 55
#define NOTE_AS1 58
#define NOTE_B1 62
#define NOTE_C2 65
#define NOTE_CS2 69
#define NOTE_D2 73
#define NOTE_DS2 78
#define NOTE_E2 82
#define NOTE_F2 87
#define NOTE_FS2 93
#define NOTE_G2 98
#define NOTE_GS2 104
#define NOTE_A2 110
#define NOTE_AS2 117
#define NOTE_B2 123
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
```

```
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE_B7 3951
```

```
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE_D8 4699
#define NOTE_DS8 4978
```

Основний код скетчу має наступний вигляд:

```
#include «pitches.h»

// ноти в мелодії:
int melody[] = {
NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4};
// тривалість ноти: 4 = четверть ноти, 8 = восьма частина,
и т.д.:
int noteDurations[] = {
4, 8, 8, 4, 4, 4, 4, 4 };
void setup () {
//перебір нот мелодії:
for (int thisNote = 0; thisNote < 8; thisNote++) {
// для розрахунку тривалості мелодії була взята одна
секунда //розділена на тип ноти.
// четверть ноти = 1000 / 4, восьма частина = 1000/8, і
т.д.. int noteDuration = 1000/noteDurations[thisNote];
tone (8, melody[thisNote],noteDuration);
//для того, щоб розділити ноти, встановимо мінімальну паузу між
ними.
// наприклад, тривалість ноти + 30% :
int pauseBetweenNotes = noteDuration * 1.30;
delay (pauseBetweenNotes);
// stop the tone
playing: noTone (8);
}
}
void loop () {
// нема потреби повторювати мелодію.
}
```

Хід виконання роботи

1. Зібрати макет відповідно завдання.
2. Підключити схему до живлення(5В).
3. Завантажити програму в мікроконтролер Arduino.
4. Перевірити правильність роботи програми.

Завдання

Зібрати схему зображену на рис.1. Створити свій унікальний набір тонів, використавши приклад, що наведений вище. Виконати додаткові завдання, відповідно до варіанту.

Парні номери:

Створити додаткову функцію, яка дозволить відтворювати певний тон (ноту), відповідно до відстані, яку зчитує дальномір (схема підключення на рис. 2). Наприклад, 5 см – нота До, 15 см – Ре і т.п. Основи роботи з дальноміром описані в практичному занятті №7.

Непарні номери:

Під'єднати 8 кнопок та запрограмувати їх, при натисканні, видавати певний тон (ноту) (одна кнопка – одна нота).

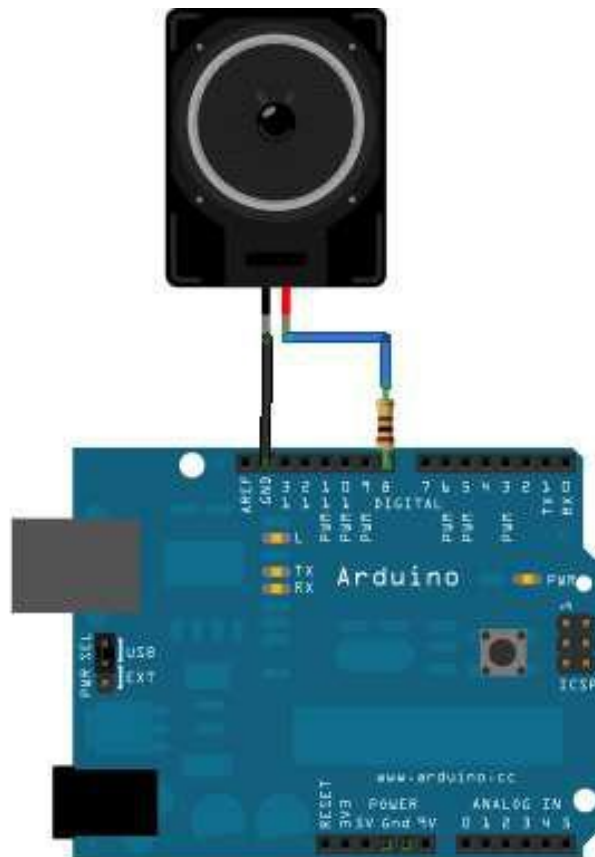


Рис. 1 – Схема підключення звукового динаміку

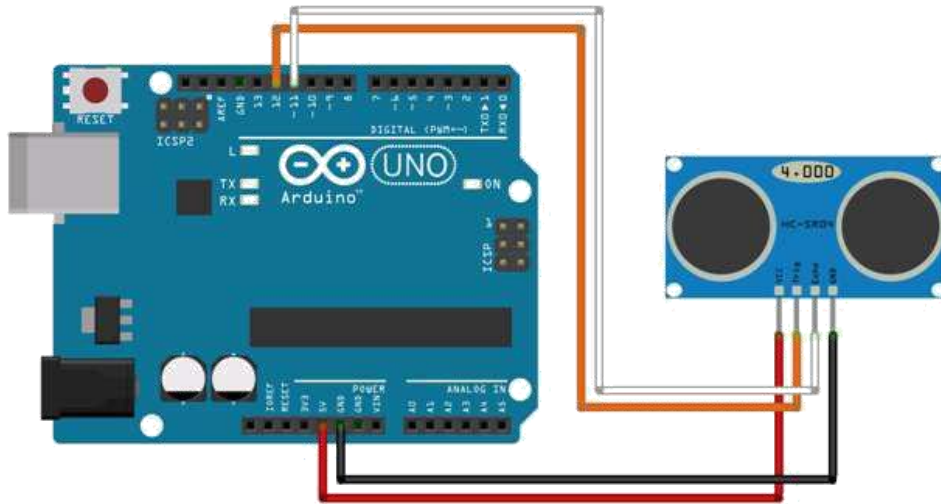


Рис. 2 – Схема підключення ультразвукового дальноміру

Підготувати звіт згідно ДСТУ 3008-95 (лістинг програми, висновки, перелік посилань).

Контрольні питання

1. Яку команду використовує Arduino для відтворення звуку? Опишіть основні її особливості.
2. Які параметри має функція *tone()*?
3. Які дані має містити файл *pitches.h* та опишіть його призначення?
4. Опишіть алгоритм програми створення мелодії, а також відтворення нот.
5. З чого складається макет роботи?