Кафедра інформатики

**КУРСОВА РОБОТА**

Тема: «Фізичні експерименти на основі Arduino»

**Виконав**:

Ліцеїст 10-А класу

Пасько Євгеній Юрійович

**Керівник:**

Чашка Юрій Михайлович

**Робота допущена до захисту:**

Чашка Юрій Михайлович

Зміст

[***Розділ 1. Вступ*** 3](#_Toc509585452)

[**Актуальність** 3](#_Toc509585453)

[**Проблематика** 3](#_Toc509585454)

[**Мета** 4](#_Toc509585455)

[***Розділ 2. Теоретична частина*** 4](#_Toc509585456)

[**Що таке звук?** 4](#_Toc509585457)

[**Чому відбувається резонанс?** 6](#_Toc509585458)

[**План проведення експерименту** 6](#_Toc509585459)

[**Класифікація звуків** 8](#_Toc509585460)

[**Використані елементи** 9](#_Toc509585461)

[***Розділ 3. Практична частина*** 10](#_Toc509585462)

[**Реалізація** 10](#_Toc509585463)

[**Особливість системи** 16](#_Toc509585464)

[**Принцип дії** 17](#_Toc509585465)

[**Транспортування даних** 18](#_Toc509585466)

[***Розділ 4. Експерименти*** 19](#_Toc509585467)

[***Висновки:*** 20](#_Toc509585468)

[***Джерела:*** 21](#_Toc509585469)

# ***Розділ 1. Вступ***

## **Актуальність**

Проведення фізичних експериментів на недорогій та розповсюдженій продукції є дуже важливою частиною навчання, зокрема в робототехніці. Цей напрямок доповнює шкільну освіту вивченням науково-технічних процесів, які поширені у сучасному суспільстві. Наукові експерименти, в свою чергу, дозволяють оволодіти навичками створення програмних і апаратних засобів, які допомагають більш змістовно зрозуміти фізичні явища.

Тема моєї роботи, а саме фізичні експерименти на основі Arduino є наразі актуальною для навчального процесу за кількох причин:

* Сприяє збільшенню навичок використання мікроконтролерної електроніки, узгодження та підключення її до персонального комп’ютеру;
* Дозволяє дослідити утворення зворотного зв’язку і утворення через це акустичних коливань. Підвищує впевненість поводження людей із використанням ними приладів;
* Продемонструвати ефективність програм автоматизації роботи приладів, та вдосконалити навички програмування на мовах програмування Arduino та C#.

## **Проблематика**

При використанні акустичних систем, не тільки на побутовому рівні але навіть і у концертних залах можливо спостерігати виникнення дискомфортного явища резонансного гудіння з динаміків.

У роботі досліджується фізичні ознаки, які впливають на параметри цього ефекту.

Зазвичай помилки у використанні не призводять до якихось руйнівних ефектів або серйозних змін, проте деякі потрібно гасити у момент виникнення небажаного ефекту і розуміти, яким чином можливо запобігати його виникненню.

Отже, однією з таких неточностей є отримання резонансу під час піднесення мікрофону заблизько до динаміків, що викликає різкий звук і негативні відчуття.

## **Мета**

Метою моєї курсової роботи є дослідити вказану вище проблему, а саме резонанс під час піднесення мікрофону заблизько до динаміків, та засвоїти навички програмування і створення пристроїв. Для вивчення даної проблеми у курсовій роботі створено пристрій та програми за допомогою Arduino у середовищі програмування Arduino IDE мовою програмування C, і у середовищі Visual Studio - мовою програмування C#.

За допомогою цього дослідження корисним було отримання інформації про конструктор Arduino та його можливостей у питанні роботи зі звуком.

У роботі також було обговорено отримання даних з порту і збереження даних у вибраній директорії.

# ***Розділ 2. Теоретична частина***

## **Що таке звук?**

*Звук* — коливальний рух частинок пружного середовища, що поширюється у вигляді хвиль у газі, рідині чи твердому тілі. Більшість явищ у природі супроводжуються характерними звуками, які сприймаються та розпізнаються вухом людини і тварин і служать для орієнтування та спілкування. Розділ науки, що вивчає звуки, називається акустикою.

Характеристиками звуку є частота, довжина хвилі, амплітуда і швидкість, а також тембр.

Найпростішим типом звуку є звук, в якому тиск у кожній точці простору змінюється за синусоїдним законом, тобто здійснює гармонічні коливання з певною частотою. *Частота* — це кількість коливань певної точки звукової хвилі в секунду. Одному циклу коливання в секунду відповідає величина 1 Гц (1/с).

Людина чує звук з частотами від 16 Гц до 20 кГц. Границі чутності визначені не строго і змінюються від людини до людини, отже цей діапазон називають чутним діапазоном. Звуки з частотами до 16 Гц називаються інфразвуком, понад 20000 Гц — ультразвуком. Звуки з частотою 109-1013 Гц називають гіперзвуком.

Людське вухо сприймає та розрізняє частоту звукових коливань як висоту звуку або тон.

## **Чому відбувається резонанс?**

Робота мікрофона задана таким чином, що коли виникає чутний звук, а саме звукові коливання, то змінюється ємкість мікрофона, а відповідно і його потенціал. Тому виникають коливання току, що надходять до динаміків і там перетворюються назад у звукові коливання. Це створює зворотній зв’язок між динаміками і мікрофоном. Проте для виникнення збудження ці обидва елементи повинні мати доступ один до одного, тобто знаходитися на невеликій відстані, направлятися один на одного та не мати перешкод для проходження звуку. За цих умов і створюється замкненість звукових хвиль у системі і з’являється неприємний свист. Це і є явище резонансу. Наявність такої проблеми часто зустрічається у виступах недосвідчених ораторів або музикантів, тож ця тема потребує вирішення.

Для розв’язання цієї задачі скористаємося методом простого відключення динаміків від мережі.

Це означає, що при отриманні з порту значення, що буде перевищувати певний вказаний поріг, то вся програма на деякий час просто відключається, а потім повертається у своє звичне становище.

## **План проведення експерименту**

Для проведення досліджень, пов’язаних із виникненням надто гучного звуку

у середовищі Visual Studio, я у декілька кроків приводжу отримані дані до правильного формату:

* Мікрофон отримує дані і перетворює їх у числові значення;
* У середовищі Arduino ці значення перетворюється у мілівольти за формулою:

int mv = analogRead(micPin) \* 5.0 / 1024.0 \* 1000.0;

* Потім отримана інформація конвертується у децибели \* 100 за формулою:

int dB = 1000 \* log10(mv);

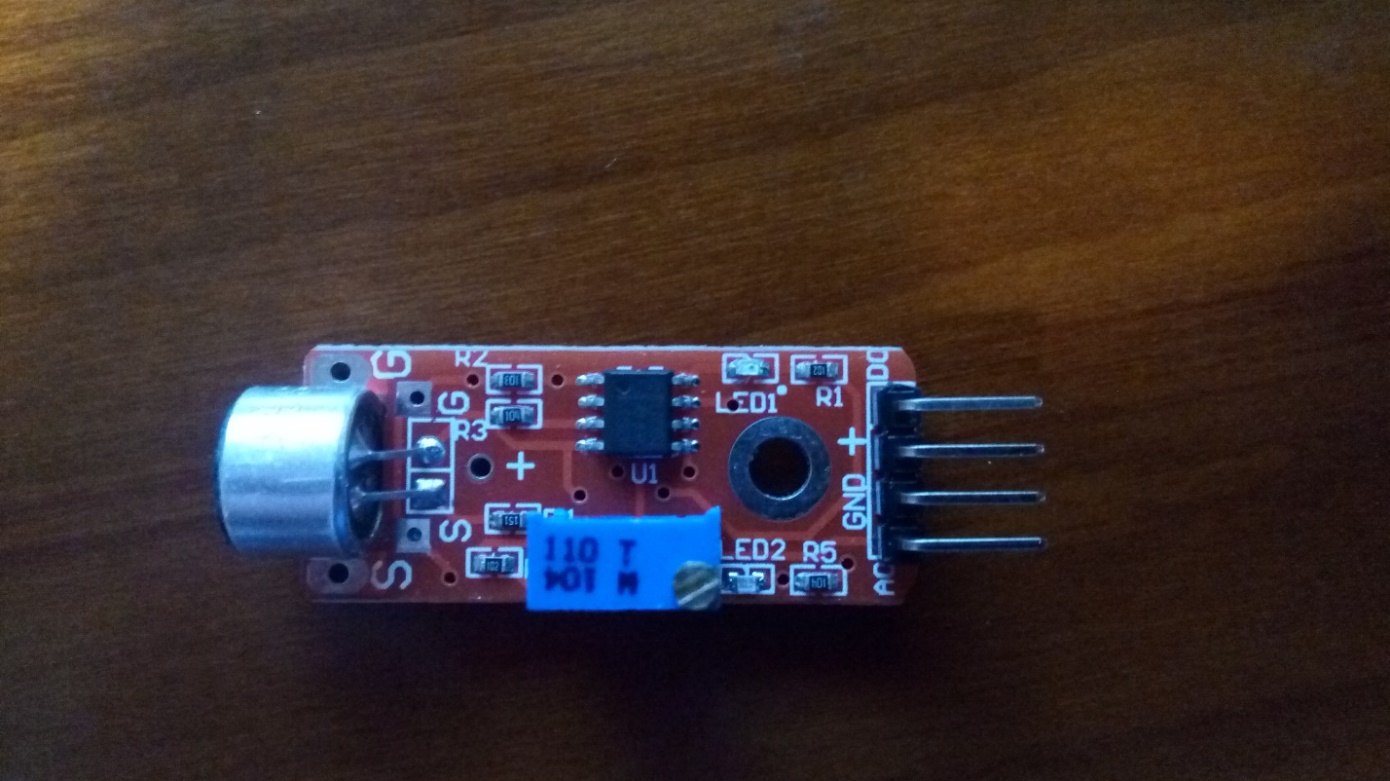
* Кінцевий результат відправляєтся на порт;
* Visual Studio приймає дані з порту та використовує у подальшому.

## **Використані елементи**

Під час даної курсової роботи я користувався такими елементами, як:

Плата Arduino Uno

Та датчик звуку



Будова мікрофона: Изображение выглядит как текст, карта

Описание создано с очень высокой степенью достоверностиИзображение выглядит как электроника, цепь

Описание создано с очень высокой степенью достоверности

# ***Розділ 3. Практична частина***

## **Реалізація**

Головною частиною та провідною думкою роботи є переведення даних, отриманих мікрофоном, через порт до самого комп’ютера для подальшого розглядання і потенційного дослідження:

private void serialPort1\_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)

{

string volume = serialPort1.ReadLine();

this.BeginInvoke(new LineReceivedEvent(LineReceived), volume);

}

private delegate void LineReceivedEvent(string volume);

private void LineReceived(string volume)

{

label1.Text = volume;

}

Для досягнення поставлених задач було використано середовище програмування Visual Studio і мову програмування С#. Частина коду наведена вище зчитує отримані з порту за допомогою мікрофона дані у необмеженій кількості, після чого перетворює їх у потрібні для подальших дій значення, тобто записує у текстовому форматі та цілочисленому, що потребується для правильного обчислення.

Реалізація кнопки запуску\скиду програми виглядає так (***1***):

private void button1\_Click(object sender, EventArgs e)

{

if (serialPort1.IsOpen == false)

{

serialPort1.Open();

serialPort1.DataReceived += serialPort1\_DataReceived;

timer1.Enabled = true;

}

else

{

chart1.Series[0].Points.Clear();

x = 0;

listBox2.Items.Clear();

listBox3.Items.Clear();

}

}

Дана частина коду відповідає за увімкнення самого отримання даних порту та таймеру запису та виводу на екран.

При повторному натисненні дані скидаються.

Другою важливою частиною є вимкнення звуку та зупинка програми за отримання зависоких значень (***2***).

private void timer1\_Tick(object sender, EventArgs e)

{

x++;

if (String.Compare(lblVolume.Text, "fatal error") == 1)

{

SendMessageW(this.Handle, WM\_APPCOMMAND, this.Handle, (IntPtr)APPCOMMAND\_VOLUME\_MUTE);

timer1.Stop();

Thread.Sleep(2000);

y = 1833;

lblVolume.Text = "1833";

timer1.Start();

SendMessageW(this.Handle, WM\_APPCOMMAND, this.Handle, (IntPtr)APPCOMMAND\_VOLUME\_DOWN);

SendMessageW(this.Handle, WM\_APPCOMMAND, this.Handle, (IntPtr)APPCOMMAND\_VOLUME\_UP);

}

else

{

y = int.Parse(lblVolume.Text);

chart1.Series[0].Points.AddXY(x, y);

Array.Resize(ref arraynumx, x);

Array.Resize(ref arraynumy, x);

arraynumx[x - 1] = x;

arraynumy[x - 1] = y;

listBox2.Items.Add(y);

listBox3.Items.Add(x + " - " + y);

}

}

Для досягнення поставлених задач було використано середовище програмування Visual Studio і мову програмування С#. Код вище перевіряє належність отриманого результату до “fatal error”, що видає порт, до якого підключений робот Arduino, програма якого видає результат “fatal error” (***3***):

void loop() {

int mv = analogRead(micPin) \* 5.0 / 1024.0 \* 1000.0; // значения в милливольтах

double dB = 1000 \* log10(mv);

if(dB > 30.0)

{

Serial.println("fatal error");

}

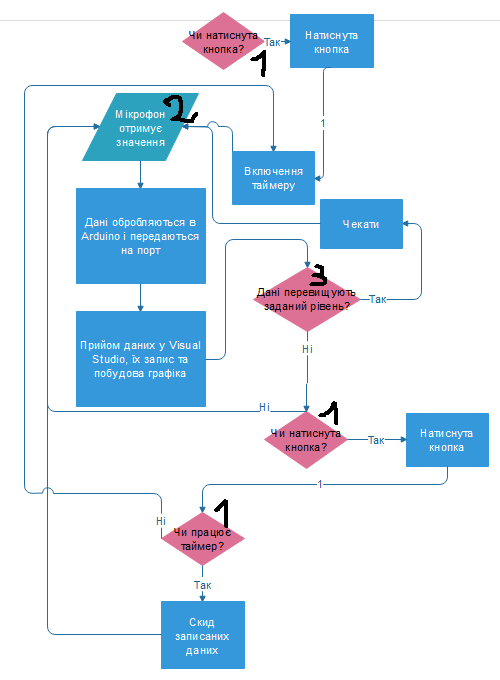
else

{

Serial.println(dB);

}

}



Під час роботи над системою було розроблено декілька додаткових можливостей, як, наприклад, можливість перегляду числових значень, отриманих системою у будь-який записаний момент часу та їх збереження у текстовому файлі у будь-якій директорії (***4***):

else

{

chart1.Series[0].Points.AddXY(x, y);

Array.Resize(ref arraynumx, x);

Array.Resize(ref arraynumy, x);

arraynumx[x - 1] = x;

arraynumy[x - 1] = y;

listBox2.Items.Add(y);

listBox3.Items.Add(x + " - " + y);

}

, та (***5***):

private void button5\_Click(object sender, EventArgs e)

{

string file\_out = "result.txt";

saveFileDialog1.FileName = file\_out;

saveFileDialog1.InitialDirectory = Application.StartupPath;

saveFileDialog1.DefaultExt = "txt";

saveFileDialog1.AddExtension = true;

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

file\_out = saveFileDialog1.FileName;

StreamWriter sw = File.CreateText(file\_out);

for(int i = 0; i < x; i++)

{

sw.WriteLine(arraynumx[i] + " " + arraynumy[i]);

}

sw.Close();

}

Останньою була додана можливість знаходження середнього значення у вказаному проміжку на listbox (***6***):

private void listBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

if(firstIndex == -1)

{

firstIndex = listBox2.SelectedIndex;

}

else

{

secondIndex = listBox2.SelectedIndex;

sum = 0;

if (firstIndex > secondIndex)

{

for (int i = secondIndex; i <= firstIndex; i++)

{

sum += arraynumy[i];

}

sredn = sum / (firstIndex - secondIndex + 1);

}

else if (firstIndex < secondIndex)

{

for (int i = firstIndex; i <= secondIndex; i++)

{

sum += arraynumy[i];

}

sredn = sum / (secondIndex - firstIndex + 1);

}

else

{

sum = arraynumy[secondIndex];

sredn = arraynumy[secondIndex];

}

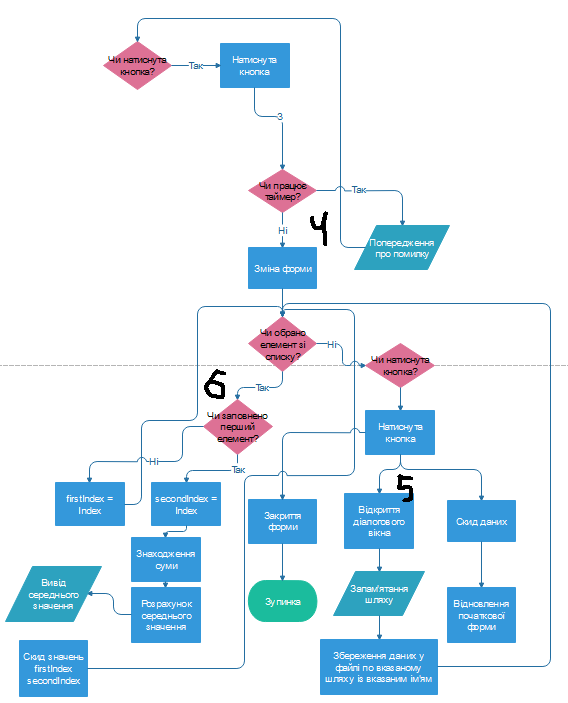
firstIndex = -1;

secondIndex = -1;

lblSredn.Text = "" + sredn;

}

}



## **Принцип дії**

1. Програма відкривається. На формі знаходяться 4 кнопки і система координат;
2. Користувач натискає кнопку запуску;
3. Програма будує графік з отриманих з мікрофона даних, а при повторному натисненні на кнопку запуску графік очищується і дані пропадають;
4. При натисненні кнопки «Стоп» програма тимчасово припиняє роботу до повторного натиснення кнопки запуску;
5. Застосування кнопки «Запис» за умови паузи в роботі мікрофона відкриває 2 списки з поточними даними, записаними у системі та з’являються нові елементи, що стосуються списків. В іншому випадку користувач бачить попередження про обов’язкове призупинення мікрофону;
6. Користувач може побачити середнє значення звуку у будь-який проміжок часу – для цього йому лише треба обрати два ключових значення списку, після чого програма вирахує середнє число з усіх значень, що знаходяться між зазначеними, включаючи їх самих;
7. Натискання кнопки «Зберегти» ініціалізує відкриття вікна збереження файлу в указаній користувачем директорії з замовленою назвою;
8. Кнопка «Скинути» видаляє поточні дані та повертає стандартний вигляд форми;
9. «Вихід» повністю закриває програму без зберігання будь-якої інформації.

# ***Розділ 4. Експерименти***

Під час виконання даної роботи мною був проведений експеримент, що спирається на визначення необхідного рівня звуку для збудження мікрофону.

Експеримент проводився таким чином, що мікрофон розташовувався на різній відстані від джерел звуку, що видавали звуки різної гучності. За допомогою цих експериментів вдалося виявити, що при працюючих динаміках реакція на звук відбувається за середньої гучності навіть за відстані 20-25 см від джерела.

Для навушників на максимальній гучності ефект сприймається лише за відстані 5 см, що є природнім, оскільки навушники назначені для передання звуку у конкретну точку, а не у простір.

# ***Висновки:***

Поставлену перед собою задачу вважаю повністю виконаною. Було розроблені програмні продукти на основі Arduino, поставлені задачі у цілому були виконані, було розглянуто програми по завантаженню даних через порт і розроблена програма на основі розглянутого методу, аналіз отриманих значень і зберігання файлів із ними у потрібному вигляді. Також була досліджена і в цілому виконана програма на методі побудови графіків. Були глибше дослідженні теми: система таймера, робота з файлами, підключення додаткових засобів. Були розглянуті теми звуку, технічних проблем і вирішення помилки із резонансом мікрофону з динаміками.

# ***Джерела:***

1. <http://pidruchniki.com/18760415/ekologiya/vpliv_zvukiv_lyudinu>
2. <http://healthy-society.com.ua/index.php?option=com_content&view=article&id=411:2011-08-03-04-46-56&catid=35:2011-04-19-08-30-36&Itemid=57>
3. <http://www.cyberforum.ru/csharp-net/thread436447.html>