




Дыбыспен жұмыс






Arduino көмегімен дыбысты генерациялаудың бірнеше жолы бар. Ең қарапайым жолы - **tone ()** функциясын қолдану. Сондай ақ, Arduino-ның негізгі платасына түйреуіш жалғағыштары арқылы қосылатын және Arduino музыкалық мүмкіндіктерін кеңейтетін әртүрлі қосымша платалар бар. Егер сізде Arduino Due платасы болса, онда дыбысты генерациялау үшін **кірістірілген цифрлық-аналогтық түрлендіргіш (ЦАТ)** жеткілікті.

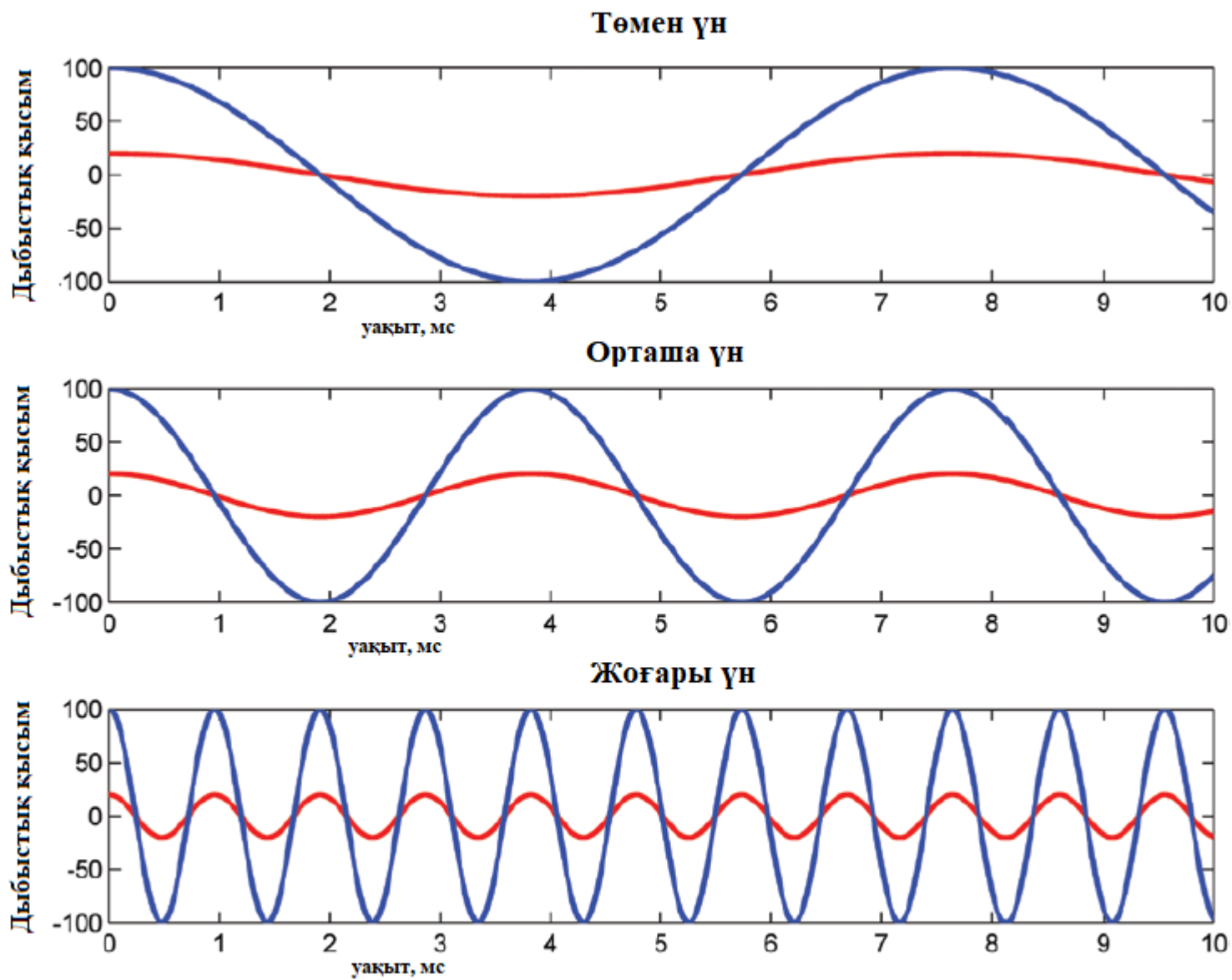
Дыбыстың қасиеттері.

Arduino көмегімен дыбысты генерация жасауды бастамас бұрын, дыбыстың не екенін және адамдар оны қалай қабылдайтынын түсініп алу қажет.


Дыбыс толқын түрінде ауа арқылы таралады. Дыбыстық динамиктердің жұмысы, барабанға немесе қоңырауға соғу ауаның дірілін тудырады. Ауаның бөлшектері тербелістердің есебінен энергияны одан әрі таратады. Қысым толқыны дыбыс көзінен, тербелмелі бөлшектердің реакциясы арқылы сіздің дабыл жарғағыңызға беріледі. Енді, осы білім Arduino платасы көмегімен дыбыстарды генерациялауға қалай көмектесетінін көрейік?




Сіз осы тербелмелі бөлшектердің екі қасиетін басқара аласыз: **жиілік және амплитуда**. **Жиілік** дегеніміз - ауа бөлшектерінің тербелісінің жылдамдығы, ал **амплитуда** - олардың тербелістерінің ауқымы. Физикалық тұрғыда үлкен амплитудалы дыбыстар, кішкентайға қарағанда жоғары болады. Жоғары жиілікті дыбыстардың үні жоғары (мысалы, сопрано), ал төмен жиілікті дыбыстар төмен (мысалы, бас). Әртүрлі амплитудасы мен жиілігі бар дыбыстық толқындар бейнеленген 1 суретті қарастырайық.



Сурет 1. Әртүрлі жиілікті және амплитудалы дыбыстық толқындар



1 суретте фортепианоның үш нотасына сәйкес келетін графиктер көрсетілген: төмен, орташа және жоғары. Мысал ретінде жиілігі 261,63 Гц болатын бірінші октаваның **До** нотасын қарастырайық. Бұл нотаны ойнату кезінде динамик, гитара немесе фортепиано секундына 261,63 тербеліс жасайтын дыбыстық толқынды генерациялайды. Толқынның тербеліс кезеңін есептеуге болады ($1/261,63 = 3,822$ мс), бұл графиктегі толық тербеліске сәйкес келеді. Arduino платасы әрбір нотаның тембрін орната отырып, **меандр** үшін кезеңді белгілеуге мүмкіндік береді. Arduino іс жүзінде нақты әлемде кең таралған синусоидалы толқынды жасай алмайды. **Меандр** – цифрлық кезеңдік сигнал болып табылады - ол екі деңгей: жоғары және төмен арасындағы жылдам ауысу (9 дәріс 1 суретті қараңыз). Нәтижесінде дыбысты беретін қысым толқыны пайда болады, бірақ дыбыс синусоидалы толқынға толық сәйкес келмейді.

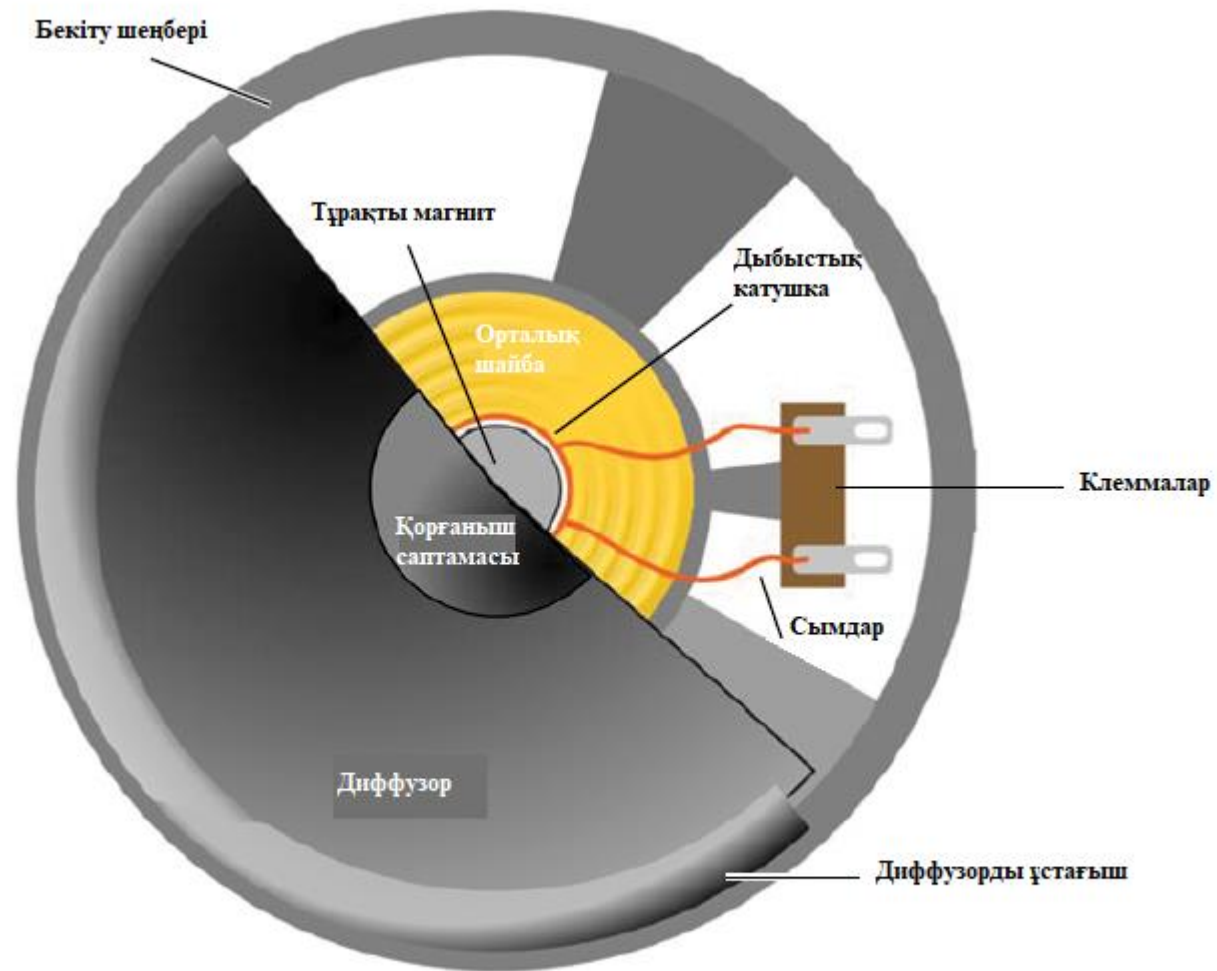


Амплитудаға келетін болсақ, оны динамик көмегімен токты өзгерту арқылы басқаруға болады. Потенциометрді динамикпен тізбектей қосу дыбыс деңгейін реттеуге мүмкіндік береді.


Динамик дыбысты қалай шығарады.

Динамиктер, өткен тақырыпта қарастырған қозғалтқыштар сияқты, электрлік сигналдарды механикалық қозғалысқа түрлендіру үшін электромагниттерді қолданады.


Динамиктің артқы жағындағы металл бөлігін мұқият зерттеңіз. Ерекше не байқадыңыз? Металл заттар оған жабысады, өйткені бұл магнит. Оны 2 суреттегі динамик құрылғысына қарап көруге болады.



Сурет 2. Динамик құрылымы



Тұрақты магниттің алдында дыбыстық катушка орналасқан. Сіз оған синусоидалы түрдегі электр сигналын берген кезде (немесе Arduino жағдайында меандр), айнымалы ток магнит өрісін жасайды, ол дыбыстық катушканың диффузорды жоғары және төмен жылжытуына алып келеді. Бұл өзара қозғалыстар диффузордың дірілдеуіне және динамиктен дыбыс шығуына алып келеді.




Дыбыстарды генерациялау үшін *tone()* функциясын қолдану
Arduino IDE-де еркін жиіліктегі дыбыстарды генерациялау
үшін арналған кірістірілген функция бар. *tone()* функциясы
берілген жиілікпен меандр қалыптастырады және оны сіз
таңдаған Arduino шығыс контактісіне жібереді.



tone() аргументтері:


- бірінші аргумент толқынды генерациялау үшін Arduino контактісінің нөмірін орнатады;
- екінші аргумент сигнал жиілігін береді;
- үшінші (міндетті емес) аргумент дыбыстың ұзақтығын анықтайды; егер бұл аргумент орнатылмаса, дыбыс *noTone()* функциясы шақырылғанға дейін жалғаса береді.



tone() функциясы ATmega контроллерінің аппараттық таймерлерінің бірімен өзара әрекеттеседі, сондықтан оны шақыруға және Arduino-мен жұмыс істеуді жалғастыруға болады, ал дыбыс фонда ойнайды. Сіз tone () функциясымен әр түрлі оқиғаларға жауап ретінде дыбыстық сигнал орната аласыз (батырмаларды басу, қашықтық датчиктерінен белгілі бір мәндерді алу, акселерометрлер және т.б.).

Тақырыпша файлын қосу.


Музыкалық дыбыстарды ойнату қажет болғанда, музыкалық ноталар үшін жиілікті анықтайтын тақырып файлын жасау пайдалы. Бұл қарапайым музыкалық әуендерді құрастыру кезінде бағдарламаны түсінікті етеді. Ноталармен таныс адамдар, ноталардың әріптермен белгіленетінін біледі. Arduino IDE-де барлық ноталар үшін жиілік мәндері бар арнайы файл бар. Оны каталогтардан іздемеңіз, тек www.exploringarduino.com/content/ch5 сайтына кіріңіз және жұмыс үстеліне жүктеңіз.



Содан кейін Arduino IDE-де жаңа бос файл құрыңыз. Сіз байқағаныңыздай, Arduino IDE сол файл атауына сәйкес буманың ішінде жаңа файл құрады. Осы бумаға жаңа файлдарды қосу арқылы сіз оларды бағдарламаңызға қоса аласыз, нәтижесінде код жақсырақ құрылымдалған болады. Жұмыс үстелінде сақталған **pitches.h** файлын көшіріңіз және жаңа жоба үшін Arduino IDE құрған бумаға сақтаңыз. Енді осы файлды Arduino IDE-де қайтадан ашыңыз. Екі қойындыға назар аударыңыз (сурет 3).



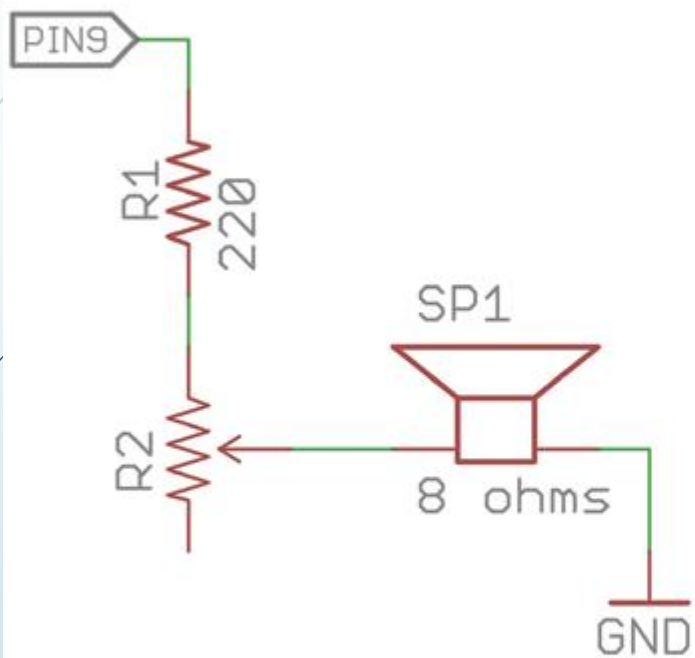
Сурет 3. Такырыпша файлдарынын екі қойындысы бар Arduino IDE терезесі



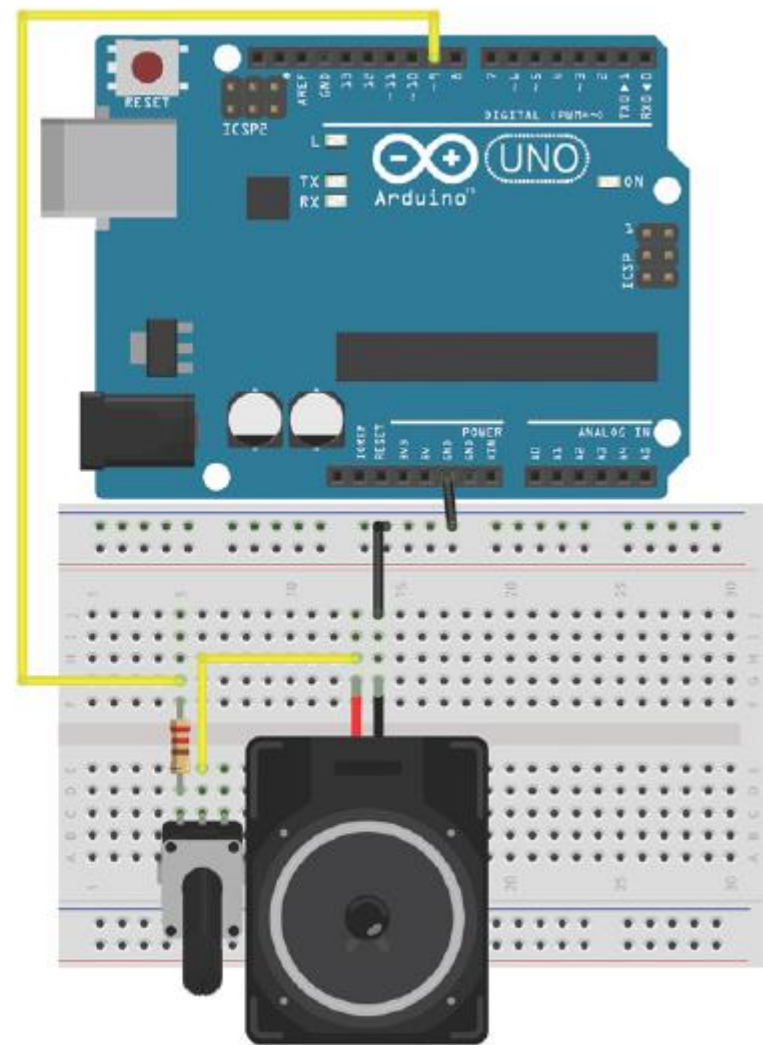
Файл мазмұнын көру үшін `pitches.h` қойындысына өтіңіз. Бұл тек нота атаулары және жиілік мәндерінің сәйкестігін орнататын анықтамалық операторлардың тізімі екендігіне назар аударыңыз. Arduino бағдарламаларды компиляциялау кезінде осы анықтамаларды пайдалану үшін, компиляторға осы файлды қайдан іздеу керектігін хабарлау керек. Мұны жасау оңай. *.ino файлының басына тиісті код қатарын қосыңыз : **`#include «pitches.h»`**.

Динамикті қосу

Енді ноталарға арналған тақырыпша файлы қосылған кезде, схеманы жинап, дыбыстарды ойнататын бағдарлама жазуға болады. Электрлік схема өте қарапайым - динамикті Arduino шығыс контактілерімен қосу керек. Алайда, қосу кезінде ток шектейтін резисторларды есте сақтау қажет.



Сурет 4. Динамикті және дыбыс реттегішті қосу схемасы



Сурет 5. Динамикті қосу схемасы

Жарықдиодтарын қосу кезіндегі секілді, динамикпен тізбектей ток шектейтін резисторды қою керек. Arduino-ның әрбір шығысы 40 мА-ден аспайтын ток шығара алады. Біздің динамиктің ішкі кедергісі – 8 Ом (сатылудағы динамиктердің көпшілігінде). Бұл электромагнитті құрайтын сым орамдарының кедергісі. Естеріңізде болса, Ом заңы $U=I*R$. Arduino шығыс кернеуі **5 В** және ток **40 мА** аспауы тиіс.

Сондықтан, минималды кедергі $R = 5 \text{ В} / 40 \text{ мА} = 125 \text{ Ом}$ болуы керек екенін анықтаймыз. Динамиктің кедергісі 8 Ом болғандықтан, токты шектейтін резистордың минималды кедергісі $125 - 8 = 117 \text{ Ом}$ болады. Резистордың ең жақын мәні-150 Ом. Кедергіні реттеу арқылы динамиктің дыбыс деңгейін өзгертуге болады. Оны резистормен тізбектей потенциометр орнатып жасай аламыз (сурет 4). Схемада R1-резистор 150 Ом, R2 - потенциометр 10 кОм.

Әуендерді құру. Массивтерді қолдану.

Әуенді жасау үшін массивтерді пайдалану өте ыңғайлы. Дыбыстар тізбегінің шығуы ағымдағы мәнді динамикке жіберумен, ноталар массивін іріктеп алудың қарапайым циклі арқылы жүзеге асады.

Массив - бір-бірімен өзара байланысқан бір типті элементтердің реттелген тізбегі. Мұндай элементтерді топтастыру – іріктеп алу үшін ерекше формат. Массивті нөмірленген тізім ретінде ұсынуға болады. Массивтің әрбір элементінде, оның тізімде орналасқан жерін көрсететін индекс бар. Біздің мысалда массивте дыбыстық тізбек сақталады – рет бойынша ойнататын ноталар тізімі.

Arduino-да массивті көрсету кезінде оның өлшемін көрсету керек. Оны массивтің өлшемін нақты көрсету арқылы немесе массивті барлық мәндермен толтыру арқылы жасауға болады. Мысалы, егер массив **int** типті төрт элементтерден тұру керек болса, біз оны бағдарламада келесідей көрсетеміз: *int numbers [4];*

Қажет болса, көрсету кезінде массивті мәндермен орнатуға болады. Массивті осылай ұсыну кезінде массив элементтерінің санын көрсету міндетті емес, массив ұзындығы көрсетілген элементтердің санына тең:

// Екі нұсқа да дұрыс көрсетілген

```
int numbers[4] = {-7, 0, 6, 234};
```

```
int numbers[] = {-7, 0, 6, 234};
```

Массивтер нөлден бастап индекстелетініне назар аударыңыз. Массив элементіне массив атауынан кейін квадрат жақшаға тиісті индекс мәнін қою арқылы қол жеткізуге болады. Мысалы, егер сіз Arduino 9 шығысына қосылған жарықдиодының жарықтығын, массивтегі үшінші элементтің мәніне тең етіп орнатқыңыз келсе, оны келесідей көрсете аласыз: `analogWrite(9, numbers[2]);`

2 индексі массивтегі үшінші мән екенін ескеріңіз, өйткені нөмірлеу нөлден басталады. Массив мәндерінің бірін келесідей өзгертуге болады: `numbers[2] = 10;`

Әрі қарай, массивтер динамикте ойнатылатын ноталар тізбегін қамтуы мүмкін құрылымды жасау үшін қажет болады.

Нота массивтерін құру және олардың айналу ұзақтығын анықтау

Ойнатқымыз келетін әуен туралы ақпаратты сақтау үшін бірдей ұзындықтағы екі массив құрыңыз. Біріншісі ноталар тізімін, екіншісі- әр нота үшін дыбыс ұзақтығының миллисекундтағы тізімін құрайды. Осы массивтердің индекстерін іріктеп алып, әуенді ойнатамыз. Екі массивтің ұзындығы бірдей (20 элемент). Кейбір ноталар нөлдік мәнмен беріледі, ол музыкалық үзілістер. Әрбір нотаның ойнау ұзақтығы екінші массивтен алынады.

Листинг 1. Әуен мысалы

```
//Нота массиві
int notes[] = {
NOTE_A4, NOTE_E3, NOTE_A4, 0,
NOTE_A4, NOTE_E3, NOTE_A4, 0,
NOTE_E4, NOTE_D4, NOTE_C4, NOTE_B4,
NOTE_A4, NOTE_B4, NOTE_C4, NOTE_D4,
NOTE_E4, NOTE_E3, NOTE_A4, 0
};
//Ноталардың ойнау ұзақтығы массиві, мс
int times[] = {
250, 250, 250, 250,
250, 250, 250, 250,
125, 125, 125, 125, 125, 125, 125, 125,
250, 250, 250, 250
};
```


Дыбысты ойнату бағдарламасын жазу.

Циклды қолданып, ноталардың мәндерін және олардың ұзақтығын массивтерден таңдаймыз және әр нотаның ойнатылуын жүзеге асырамыз. Әуенді қайта-қайта тыңдау қажеті болмағандықтан, циклді **setup ()** функциясында орындауға болады. Ойнатуды жалғастыру үшін Reset батырмасын басыңыз. 2 листингте Arduino ойнатқышының толық бағдарламасы беріледі.

Листинг 2. Әуендерді ойнату

```
//Динамикте әуенді ойнату
#include "pitches.h" //нота жиілігінің мәндері бар тақырыпша файлы
const int SPEAKER=9; //Динамикті қосу шығысы
//Нота массиві
int notes[] = {
NOTE_A4, NOTE_E3, NOTE_A4, 0,
NOTE_A4, NOTE_E3, NOTE_A4, 0,
NOTE_E4, NOTE_D4, NOTE_C4, NOTE_B4, NOTE_A4, NOTE_B4,
NOTE_C4, NOTE_D4,
NOTE_E4, NOTE_E3, NOTE_A4, 0
};
//Ноталардың ойнау ұзақтығы массиві, мс
int times[] = {
250, 250, 250, 250,
250, 250, 250, 250,
125, 125, 125, 125, 125, 125, 125, 125,
250, 250, 250, 250
};
void setup()
{
//Әрбір нотаны ноталар массивінен іріктеп алу
for (int i = 0; i < 20; i++)
{
tone(SPEAKER, notes[i], times[i]);
delay(times[i]);
}
}
void loop()
{
//Ойнауды қайталау үшін, reset батырмасын басу керек.
}
```