

ARDUINO TARAQQIYOTI

Ma'ruza #6

Mavzu: Arduino yordamida ovoz (tovush) hosil qilish

Reja:

1. Ovozning xususiyatlari
2. Karnay qanday ovoz chiqaradi ?
3. Ovoz hosil qilish uchun tone() dan foydalanish
4. Ta'rif faylini o'z ichiga olish
5. Karnay simini ulash
6. Ovoz ketma - ketligini yaratish
7. Nota va davomiylik massivlarini tuzish

Xulosa

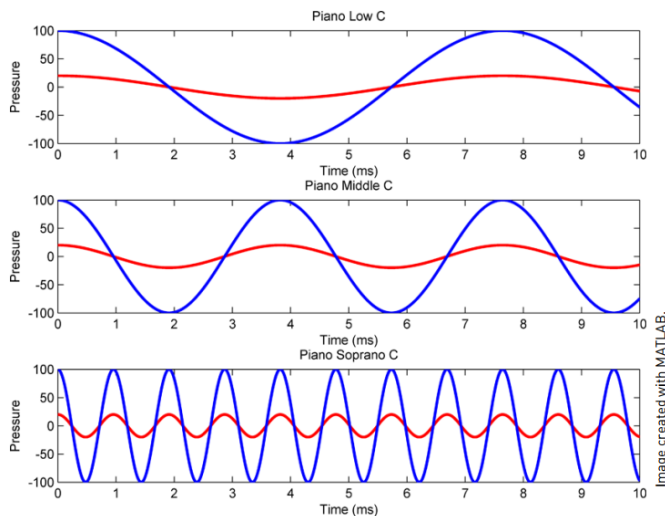
Arduinoda ovoz chiqarishni boshlashdan oldin, tovushlar nima ekanligini va odamlar ularni qanday qabul qilishini tushunishingiz kerak. Bu birinchi bo'limda siz tovush to'lqinlari qanday paydo bo'lishi, ularning xususiyatlari va bu xususiyatlarning manipulyatsiyasi musiqa, ovozlari va boshqalarni ishlab chiqarishi haqida bilib olasiz .

Ovozning xususiyatlari

Ovoz bosim to'lqini sifatida havo orqali uzatiladi. Ob'ekti sifatida kalonka, baraban, yoki qo'ng'iroq tebranadi va natijada o'zining atrofidagi havoni ham tebratadi. Havo zarralari tebranar ekan, ular energiyani atrofdagi zarrachalarga o'tkazadi va bu zarrachalarni ham tebratadi. Bu usulda, tebranish zarrachalarining zanjirli reaksiyasini yaratish orqali bosim to'lqini manbadan sizning quloq pardangizga o'tkaziladi. Xo'sh, nima uchun Arduinoda ovoz chiqarishni tushunish uchun buni bilish kerak ?

Siz bu tebranuvchi zarrachalarning ikkita xususiyatini Arduino yordamida boshqarishingiz mumkin: chastota va amplituda. Chastota havo zarralari oldinga va orqaga qanchalik tez tebranishini, amplitudasi esa ularning tebranishlarining kattaligini ifodalaydi. Fizik ma'noda yuqori amplitudali tovushlar balandroq, pastroq amplitudali tovushlar esa sokinroq. Yuqori chastotali tovushlar balandroq (soprano kabi), past chastotali tovushlar esa pastroq (bas

kabi). Har xil amplitudali va chastotali tovush to'liqlarining sinusoidal tasvirlarini ko'rsatadigan quyidagi rasmdagi diagrammani ko'rib chiqing .



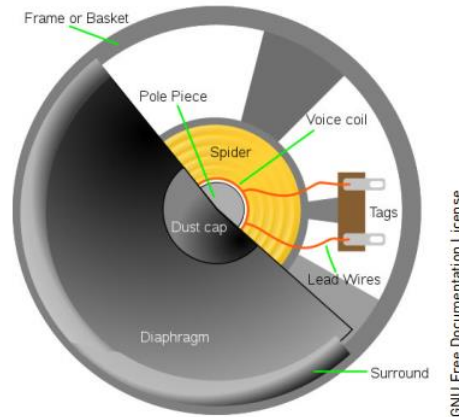
Ushbu rasmda uchta pianino notasi ko'rsatilgan: past, o'rta va soprano C. Har biri berilgan chastotalarni past va yuqori amplitudada ko'rsatadi. Misol tariqasida, chastota va amplitudani tushunish uchun o'rta C ga e'tibor qarating. O'rta C 261.63 Hertz (Gts) chastotaga ega. Boshqacha qilib aytganda, karnay, gitara tori yoki pianino tori sekundiga 261,63 tebranishni bajaradi. Bu qiymatni qaytarib, siz to'liq davrini topishingiz mumkin, uni yuqoridagi rasmda ko'rish oson. $1/261.63$ 3.822 millisekundga teng, bu grafikdagi bitta to'liq tebranishning kengligi. Arduinodan foydalanib, siz bu davrni kvadrat to'liq uchun belgilashingiz va shu bilan notaning ohangini sozlashingiz mumkin.

Muhimi, Arduino aslida siz real dunyoda kuzatishingiz mumkin bo'lgan sinusoidal to'liq qila olmaydi . Kvadrat to'liq - bu raqamli davriy to'liq - u yuqori va past qiymatlar o'rtasida ham tebranadi , lekin u sinus to'liq kabi sekin emas, bir zumda o'zgaradi. Bu hali ham tovushga olib keladigan bosim to'liqini yaratadi, lekin u sinusoidal to'liq kabi "chiroyli" emas .

Amplitudaga kelsak, siz buni dinamik orqali o'tishga ruxsat berilgan oqim miqdorini o'zgartirish orqali boshqarishingiz mumkin . Karnay bilan bir qatorda potentsiometr yordamida siz dinamikning ovoz balandligini dinamik ravishda sozlashingiz mumkin.

Karnay qanday ovoz chiqaradi ?

Dinamikalar, xuddi oldingi bobda o'rgangan dvigatellar kabi, elektromagnit kuchlardan foydalanib, elektrni harakatga aylantiradi. Karnayning orqa tomonida metall bo'lakni ushlab turishga harakat qiling. Qiziqarli narsani payqadingizmi? Metall, ehtimol, karnayning orqa qismiga yopishadi, chunki hamma dinamiklarning orqa tomoniga doimiy magnit o'rnatilgan. Quyidagi rasmda umumiy karnayning kesimi ko'rsatilgan.



Doimiy magnit rasmda ko'rsatilgan ovozli g'altak va qutb qismining orqasiga o'rnatiladi. G'altakning uchlariga sinusoidal kuchlanish signalini (yoki Arduino holatida kvadrat to'lqinni) yuborganingizda, o'zgaruvchan tok magnit maydonini qo'zg'atadi, bu esa qutb bo'lari va diafragmani doimiy magnit kabi yuqoriga va pastga tebranishiga olib keladi. Bu oldinga va orqaga tebranish, o'z navbatida, karnay oldidagi havoni tebratib, quloq pardangizga boradigan tovush to'lqinini samarali hosil qiladi.

Ovoz hosil qilish uchun tone() dan foydalanish

Arduino IDE ixtiyoriy chastotalar tovushlarini osongina chiqarish uchun o'rnatilgan funktsiyani o'z ichiga oladi. Tone() funktsiyasi siz tanlagan chiqish pinida tanlangan chastotaning kvadrat to'lqinini hosil qiladi. Tone() funktsiyasi uchta argumentni qabul qiladi, lekin oxirgisi ixtiyoriy:

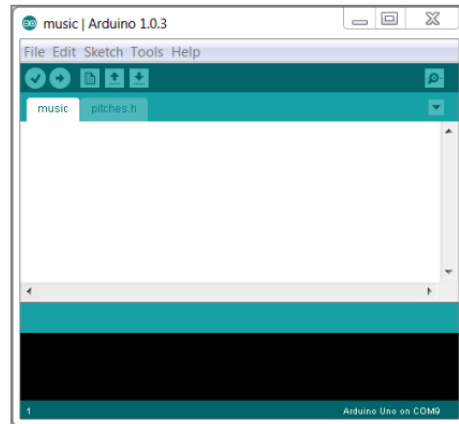
- Birinchi argument pinni ohangni ochish uchun o'rnatadi.
- Ikkinchi argument ohang chastotasini belgilaydi.
- Uchinchi (ixtiyoriy) argument ohang davomiyligini belgilaydi. Agar uchinchi argument o'rnatilmagan bo'lsa, siz noTone () ga qo'ng'iroq qilmaguningizcha ohang ijro etishda davom etadi.

Tone() ATMeganing apparat taymerlaridan birini ishlatgani uchun, siz fonni ovoz chiqarishda davom etayotgan paytda Arduino bilan ohangni boshlashingiz va boshqa narsalarni qilishingiz mumkin.

Ta'rif faylini o'z ichiga olish

Musiqani ijro etish haqida gap ketganda, nomlarni qayd etish uchun chastotalarni xaritaga keltiruvchi ta'rif fayli foydali bo'ladi. Bu oddiy musiqiy kliplarni ijro etishni yanada intuitiv qiladi. Nota musiqasini o'qishni yaxshi biladiganlar uchun, notalar ularning tovushini ifodalovchi harflar bilan ko'rsatilganligini bilasiz. Arduino IDE ushbu yozuvlarning har birini tegishli chastotaga mos keladigan sarlavha faylini o'z ichiga oladi.

Keyin Arduino IDEni oching va IDEni ochganingizda avtomatik ravishda yaratilgan bo'sh loyihani saqlang . Siz allaqachon payqaganingizdek, loyihani saqlaganingizda, u aslida shu nomli papkani saqlaydi va .ino faylini shu papkaning ichiga joylashtiradi. Bu papkaga boshqa fayllarni qo'shib, siz ularni dasturingizga kiritishingiz mumkin, bunda sizning kodingiz yaxshiroq tartibga solingan. Ish stoliga saqlagan pitches.h faylini IDE tomonidan yaratilgan papkaga nusxalash; keyin Arduino IDE -ni yoping . Arduino IDE da .ino faylini oching va hozirda paydo bo'ladigan ikkita yorliqqa e'tibor bering.



Fayl tarkibini ko'rish uchun pitches.h yorlig'ini bosing. E'tibor bering, bu odamlar o'qiy oladigan nomlarni berilgan chastota qiymatlariga mos keladigan ta'riflar ro'yxati. Faqat IDEda sarlavha faylining mavjudligi yetarli emas. Ushbu kod qatorini .ino faylining yuqori qismiga qo'shing:

```
#include "pitches.h" // Sarlavha fayli aniqliklari bilan
```

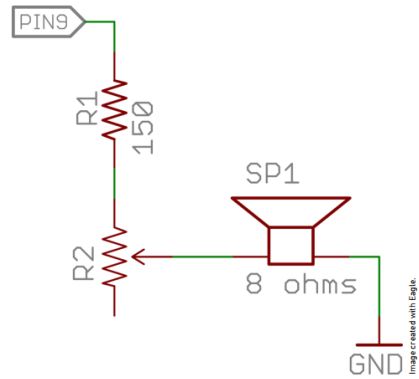
Kompilyator uchun bu, asosan, sarlavha faylining tarkibini asosiy faylingizning yuqori qismiga nusxalash va joylashtirish bilan bir xil. Biroq, bu faylni yanada tushunarli qiladi va o'qishni osonlashtiradi.

Karnay simini ulash

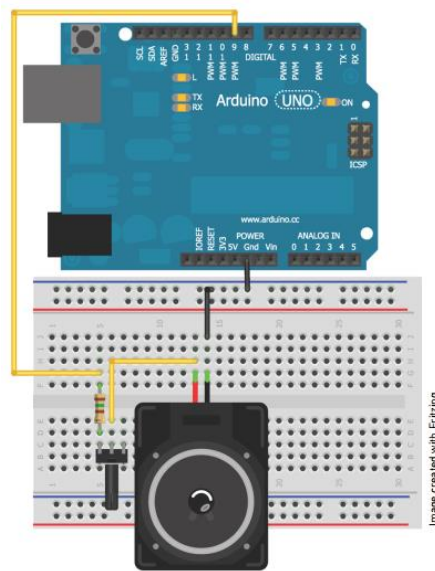
Endi sizda sarlavha faylingiz bor, siz test sxemasini tuzishga va musiqa ijro etadigan oddiy dastur yozishga tayyorsiz. Elektr sozlamalari juda oddiy va dinamikni Arduino chiqish piniga ulashni o'z ichiga oladi. Biroq, oldingi darslardan oqim cheklovli rezistorlar haqida bilganlaringizni eslang.

Xuddi LEDlarda bo'lgani kabi, siz Arduinoning kirish-chiqish pinlaridan biridan juda ko'p oqim olmasligingiz uchun siz karnay bilan bir qatorda oqim cheklovchisini qo'yishni bajarasiz . Siz ilgari bilib olganingizdek, har bir kirish -chiqish pini maksimal 40mA tok kuchiga ega bo'lishi mumkin , shuning uchun siz undan oshib ketishiga to'sqinlik qiladigan resistor qo'ying. Odatiy Arduino to'plamlari bilan keladigan karnayning ichki qarshiligi 8Ω (siz sotib oladigan ko'pchilik karnaylar kabi); Bu qarshilik elektromagnitni tashkil etuvchi simlarning o'rashidan kelib chiqadi. Eslatib o'tamiz, Ohm qonuni $V = IR$ ekanligini bildiradi. Ushbu senariyda, kirish-chiqish pindan 5V chiqadi va siz tok kuch 40mA dan oshmasligini ta'minlashingiz zarur . R

uchun echim topsangiz, minimal qarshilik quyidagicha bo'lishi kerak: $R = 5V / 40mA = 125\Omega$. 8Ω allaqachon karnay tomonidan hisoblangan, shuning uchun sizning rezistoringiz kamida $125\Omega - 8\Omega = 117\Omega$ bo'lishi kerak. Eng yaqin umumiy qarshilik 150Ω , shuning uchun siz undan foydalanishingiz mumkin. Ushbu rezistor qiymatini sozlash orqali siz karnay ovozini o'zgartirishingiz mumkin. Buni iloji boricha osonroq qilish uchun 5-4-rasmda ko'rsatilgandek, 150Ω rezistorli potentsiometrni ishlatishingiz mumkin. Sxemada R1 - 150Ω qarshilik, R2 - potentsiometr.



E'tibor bering, potentsiometrlarning oldingi ishlatilishidan farqli o'laroq, bu konfiguratsiya faqat ikkita pindan foydalanadi: o'rta (yoki siljувchi) pini karnayga o'tadi va oxirgi pinlardan biri 150Ω rezistorga ulanadi. Tugma ulanmagan terminal tomon burilgach, potentsiometrning butun qarshiligi 150Ω rezistorning ketma -ket qarshiligiga qo'shiladi va ovoz pasayadi. Qachonki tugmachani ulangan so'nggi terminal tomon burs, u ketma -ket qarshilik ko'rsatmaydi va karnay maksimal ovoz balandligida bo'ladi. Yuqoridagi sxemaning yig'ilgan ko'rinishi quyidagi rasmda keltirilgan.



Karnaylarda qutb yo'q; Siz ularni har qanday yo'nalishda ulashingiz mumkin. Karnayni muvaffaqiyatli ulashdan so'ng siz musiqa yaratishga tayyorsiz!

Ovoz ketma - ketligini yaratish

Ba'zi qo'shiqlarni ijro etish uchun siz avval bir nechta qiymatlarni osongina saqlash uchun massivlardan foydalanishni o'rganasiz . So'ngra, massivlarni takrorlash va ularni karnayda ijro etish uchun oddiy tsiklni amalga oshirasiz .

Massivlardan foydalanish

Massiv - bu qandaydir tarzda bog'liq bo'lgan qiymatlar ketma -ketligi. Ularni guruhlariga birlashtirish orqali, uni takrorlash uchun ideal format bo'ladi. Siz qatorni raqamlangan ro'yxat deb o'ylashingiz mumkin . Har bir pozitsiyada ro'yxatda o'z o'rnini ko'rsatadigan indeks mavjud va har bir indeks saqlamoqchi bo'lgan qiymatga ega. Siz bu yerda o'ynatmoqchi bo'lgan notalar ro'yxatini xohlagan tartibda saqlash uchun qatordan foydalanasiz .

Arduino xotirasi to'g'ri boshqarilishini ta'minlash uchun ma'lum uzunlikdagi massivlarni e'lon qilish kerak . Siz buni elementlar sonini aniq ko'rsatib yoki o'zingizni qiziqtirgan barcha qiymatlar bilan to'ldirish orqali qilishingiz mumkin. Masalan, agar siz to'rtta butun sonli qatorni yaratmoqchi bo'lsangiz, uni shunday yaratishingiz mumkin:

```
int raqamlar [4];
```

Siz ixtiyoriy ravishda uni e'lon qilganingizda o'zgaruvchilarga boshlang'ich qiymatlarni berishingiz mumkin. Agar siz boshlang'ich qiymatlarni ishga tushirsangiz, qavs ichida uzunlikni belgilash ixtiyoriydir. Agar aniqlanmagan bo'lsa, uzunlik siz kiritgan raqam elementlariga teng deb hisoblanadi:

```
// Bu ikkalasi ham qabul qilinadi
```

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

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

E'tibor bering, massivlar nol indekslanadi. Boshqacha qilib aytganda, birinchi raqam 0-pozitsiyada, ikkinchisi 1 -pozitsiyada va hokazo. Tegishli qiymat indeksini o'zgaruvchi nomidan keyin kvadrat qavs ichiga qo'yib, istalgan indeksdagi massiv elementlariga kirishingiz mumkin. Agar siz 9 -pinga ulangan LEDning yorqinligini qatorning uchinchi yozuviga o'rnatmoqchi bo'lsangiz, buni shunday qilishingiz mumkin :

```
analogWrite (9, raqamlar [2]);
```

E'tibor bering, raqamlash noldan boshlanganligi sababli, 2 indeksi qatorning uchinchi qiymatini ifodalaydi. Agar siz qator qiymatlaridan birini o'zgartirmoqchi bo'lsangiz , xuddi shunday tarzda qilishingiz mumkin :

```
raqamlar [2] = 10;
```

Keyinchalik, siz karnayda o'ynatmoqchi bo'lgan yozuvlar ketma-ketligini saqlaydigan tuzilmani yaratish uchun (bu misollarda ko'rsatilgandek) massivlardan foydalanasiz .

Nota va davomiylik massivlarini tuzish

Siz ijro qilmoqchi bo'lgan ohang haqidagi ma'lumotlarni saqlash uchun siz bir xil uzunlikdagi ikkita massivdan foydalanishingiz mumkin . Birinchisida maydonlar ro'yxati, ikkinchisida har bir nota millisekundlarda o'ynashi kerak bo'lgan vaqtlar ro'yxati mavjud. Keyin siz ushbu massivlarning indekslarini takrorlashingiz va ohangni ijro etishingiz mumkin .

O'rta maktabda musiqa darslarida saqlagan musiqiy mahoratimdan foydalanib , men qisqa va jozibali ohang yig'dim:

```
//Note massivi
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
};
//Har bir notaning davomiyligi (millisekundlarda)
int times[] = {
250, 250, 250, 250,
250, 250, 250, 250,
125, 125, 125, 125, 125, 125, 125, 125,
250, 250, 250, 250
};
```

E'tibor bering, ikkala massivning uzunligi bir xil: 20 ta element. E'tibor bering, ba'zi notalar 0 sifatida ko'rsatilgan. Bu musiqiy damlar (ijro etilmagan zarbalar). Har bir nota ikkinchi massivdagi davomiyligi bilan juftlashadi .

Birinchidan, berilgan notalar ketma - ketligini sinab ko'ring; keyin o'zingiz yaratishga harakat qiling!

Dasturni yakunlash

Oxirgi qadam - loyihaga ijro etish funksiyasini qo'shish. Buni massivdagi har bir indeks orqali o'tadigan va berilgan notani berilgan vaqt davomida bajaradigan oddiy for loop yordamida bajarish mumkin . Siz buni qayta -qayta eshitishni xohlamasangiz uchun, ijro etish funksiyasini setup () funktsiyasiga qo'yishingiz mumkin, shunda u faqat bir marta bo'ladi. Qayta tiklash tugmachasini bosib, ijro etishni qayta boshlashingiz mumkin. Quyidagi dastur to'liq ijro etish dasturini ko'rsatadi.

Arduino musiqa pleyeri-music.ino

```
//Karnayda ohang ijro etish
#include "pitches.h"
```

```

const int SPEAKER=9; //Karnay pini
//Notalar massivi
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
};
//Har bir notaning davomiyligi (millisekundlarda)
int times[] = {
250, 250, 250, 250,
250, 250, 250, 250,
125, 125, 125, 125, 125, 125, 125, 125,
250, 250, 250, 250
};
void setup()
{
//Har bir notani munosib davomiylilik uchun ijro etish
for (int i = 0; i < 20; i++)
{
tone(SPEAKER, notes[i], times[i]);
delay(times[i]);
}
}
void loop()
{
}

```

Agar siz o'z musiqangizni yaratmoqchi bo'lsangiz, massivlar teng uzunlikda ekanligiga ishonch hosil qiling va for () siklining yuqori chegarasini o'zgartiring. Chunki tone () funktsiyasi fonda ishlatilishi mumkin, u delay () funktsiyasidan foydalanish muhim. Kodni notaning davomiyligiga teng vaqtga kechiktirib, siz ko'rsatgan vaqt mobaynida avvalgi ijro tugamaguncha, Arduino keyingi yozuvni ijro etmasligini ta'minlang.

Tone() funktsiyasining cheklanishlarini tushunish

Tone () funktsiyasida ma'lum cheklovlar mavjud. Servo kutubxonasi kabi, tone () ham taymer tayanchiga tayanadi, u ham kartaning puls kengligi modulyatsiyasi (PWM) funktsiyasida ishlatiladi. Agar tone () dan foydalansangiz, PWM 3 va 11 -pinlarda to'g'ri ishlamaydi (Mega - dan boshqalarda).

Shuni ham yodda tutingki, Arduino I/O pinlari raqamli-analogli konvertorlar emas. Shunday qilib, ular sinus to'lqin emas, balki berilgan chastotada faqat kvadrat to'lqin chiqaradi. Garchi bu karnay bilan ohang yaratish uchun etarli bo'lsada, siz musiqani ijro etish uchun keraksiz deb topasiz.

Oxirgi cheklov shundaki, ohang () funktsiyasini bir vaqtning o'zida faqat bitta pinda ishlatishingiz mumkin, shuning uchun u bir nechta karnay boshqarish uchun ideal emas.

Xulosa

Ushbu darsda siz quyidagilar haqida bilib oldingiz:

- ■ Karnaylar havo to'lqinlari orqali o'tadigan va sizning quloqlaringiz tomonidan tovush sifatida qabul qilinadigan bosim to'lqinini yaratadilar.
- ■ Elektr tokining o'zgarishi magnit maydonini qo'zg'atadi , uni karnaydan ovoz chiqarish uchun ishlatish mumkin .
- ■ Tone() funksiyasidan ixtiyoriy chastotalar va davomiylik tovushlarini chiqarish uchun foydalanish mumkin.
- ■ Arduino dasturlash tili ma'lumotlar ketma - ketligi orqali takrorlanish uchun massivlardan foydalanishni qo'llab -quvvatlaydi.
- ■ Karnay balandligi dinamik bilan ketma -ket potentsiometr yordamida sozlanishi mumkin.

Foydalanilgan Adabiyotlar

- Jeremy Blum, “Exploring arduino: Tools and techniques for engineering wizardry”, 2013
- Electronics: A Systems Approach (6th edition), Neil Storey, Pearson Education UK, 2017
- Arduino.cc veb sayti ma'lumotlari