

# Sintetizator de Voce

- Nume: **Ioniță Alexandru-Darius**
- Grupă: **332CA**

## Introducere

- Proiectul constă în realizarea unui sintetizator de voce pe fundalul unei mașini de tip karaoke, bazat pe microcontroller-ul ATmega328P.
- Ideea proiectului a pornit în urma influenței albumului **brat** al artistei Charli XCX și a popularizării sintetizatoarelor de voce în muzica anilor 2020.
- Aceste tendințe au evidențiat interesul pentru efecte vocale artificiale și procesare digitală a vocii, inspirând dezvoltarea unui sistem hardware capabil să reproducă un astfel de efect într-o formă simplificată.

## Descriere generală

- Scopul proiectului este de a transforma vocea umană într-un semnal artificial de tip „square tooth”, generând un efect robotic controlat. Sistemul urmărește să corecteze automat notele cântate astfel încât acestea să se încadreze în cheia melodiei, eliminând dependența de acuratețea interpretării vocale.
- Utilizatorul poate selecta o melodie folosind o interfață simplă cu butoane, iar versurile sunt afișate pe un calculator. În același timp, semnalul vocal captat printr-un microfon este procesat în timp real, fiind detectată înălțimea sunetului și corectată conform tonalității melodiei selectate.
- Rezultatul este redat sub forma unui semnal de tip square wave, cu un caracter artificial, specific efectelor moderne de auto-tune.
- Proiectul este util din mai multe perspective:
  - Pentru utilizatori, oferă o experiență interactivă de karaoke în care aptitudinile muzicale nu influențează semnificativ rezultatul, deoarece toate notele sunt ajustate automat în cheia melodiei.
  - Proiectul reprezintă, de asemenea, un exercițiu de integrare hardware-software, implicând procesare de semnal, lucrul cu periferice (ADC, SPI, USART) și generarea de semnale audio. El poate constitui o bază pentru dezvoltarea ulterioară a unor sisteme mai avansate de procesare audio în timp real.

Schema bloc:



# Hardware Design

## Microfon

Nume Pin	Semnal / Functie	Pin ATmega328P
OUT	Output analog audio	PC0 (ADC0)
VCC	Supply putere	5V
GND	Ground	GND

## Modul MicroSD

Nume Pin	Semnal / Functie	Pin ATmega328P
VCC	Supply putere	5V
GND	Masa	GND
MISO	SPI Master In Slave Out	PB4
MOSI	SPI Master Out Slave In	PB3
SCK	SPI Clock	PB5
CS	SPI Chip Select	PB2

## Amplificator audio

Nume Pin	Semnal / Functie	Pin ATmega328P
IN	Input audio	PD6
VCC	Supply putere	5V
GND	Masa	GND

## Butoane

Buton	Semnal / Functie	Pin ATmega328P
Buton 1	Buton Next	PB1
Buton 2	Buton Previous	PB0
Buton 3	Buton Play	PD7

## Buzzer pasiv

Nume Pin	Semnal / Functie	Pin ATmega328P
I/O	PWM / digital output	PD5
VCC	Power supply	5V
GND	Ground	GND

# Software Design

Implementarea software a proiectului se bazează pe prelucrarea în timp real a semnalului audio recepționat de la un microfon și pe generarea unei frecvențe corectate muzical prin intermediul unui buzzer pasiv. Sistemul funcționează pe microcontrolerul ATmega328P Xplained Mini și utilizează resursele interne ale acestuia pentru achiziția semnalului, analiza frecvenței și generarea semnalului audio de ieșire.

Procesul software este împărțit în mai multe etape principale:

## 1. Obținerea semnalului audio

Semnalul audio este captat cu ajutorul unui modul microfon conectat la pinul PC0 al microcontrolerului, corespunzător canalului analogic A0. Microfonul transformă undele sonore în variații de tensiune electrică.

Microcontrolerul utilizează convertorul analog-digital (ADC - Analog to Digital Converter) integrat pentru a eșantiona semnalul audio la intervale regulate de timp. În urma acestui proces se obține o succesiune de valori numerice ce reprezintă amplitudinea semnalului audio în funcție de timp.

Datele rezultate sunt stocate într-un buffer și sunt utilizate ulterior pentru analiza frecvenței fundamentale.

## 2. Determinarea frecvenței folosind algoritmul YIN

Pentru identificarea notei muzicale cântate sau detectate de microfon se utilizează algoritmul YIN, un algoritm eficient pentru estimarea frecvenței fundamentale a unui semnal periodic.

Algoritmul funcționează prin compararea semnalului cu versiuni decalate ale acestuia și determinarea perioadei dominante. Principalele etape ale algoritmului sunt:

- calculul funcției de diferență
- normalizarea cumulativă a diferenței
- identificarea minimului local relevant
- determinarea perioadei semnalului
- conversia perioadei în frecvență

Frecvența este calculată utilizând relația:

$$f = fs / T$$

unde:

- $f$  reprezintă frecvența semnalului
- $fs$  reprezintă frecvența de eșantionare

- T reprezintă perioada detectată de algoritm

Rezultatul obținut reprezintă nota reală emisă de utilizator.

### 3. Corectarea tonală („autotune”)

După determinarea frecvenței fundamentale, sistemul compară valoarea detectată cu un set predefinit de note muzicale aparținând unei game selectate.

Fiecare notă are asociată o frecvență standard conform sistemului temperării egale. Software-ul determină cea mai apropiată notă validă din gama activă prin calculul diferenței dintre frecvența detectată și frecvențele notelor disponibile.

Exemplu: dacă utilizatorul cântă o frecvență apropiată de Fa#, iar gama selectată nu conține această notă, sistemul o poate corecta către Sol sau Fa, în funcție de regula implementată.

Această etapă reprezintă funcția principală de autotune în timp real.

### 4. Generarea frecvenței corectate

După alegerea notei corectate, microcontrolerul generează un semnal dreptunghiular cu frecvența corespunzătoare notei respective.

Pentru acest lucru este utilizat unul dintre temporizatoarele hardware interne ale microcontrolerului ATmega328P. Timerul este configurat astfel încât să producă o undă PWM sau un semnal periodic la frecvența dorită.

### 5. Redarea notei prin buzzer

Semnalul generat de timer este transmis către un buzzer pasiv conectat la unul dintre pinii digitali ai microcontrolerului.

Buzzerul convertește semnalul electric periodic în vibrații mecanice, producând astfel sunetul notei corectate.

În acest mod, utilizatorul aude în timp real varianta corectată tonal a notei detectate de microfon.

### 6. Funcționarea în timp real

Întregul proces:

- captarea semnalului
- analiza frecvenței
- corectarea tonală
- generarea sunetului

este executat continuu într-o buclă principală de program, ceea ce permite funcționarea sistemului în timp real.

Pentru reducerea latenței:

- dimensiunea bufferului este menținută relativ mică
- calculele sunt optimizate pentru arhitectura AVR pe 8 biți
- se utilizează periferice hardware interne ale microcontrolerului

## Rezultate Obținute

Rezultatele obținute în urma realizării proiectului

## Concluzii

Concluzii

- Urasc cititoarele microSD

## Download

Download

## Jurnal

Secțiune de jurnal.

Este ziua a doua când mă bat cu cititorul de card microSD și cititorul de card microSD câștigă

## Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:  
<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:  
[http://ocw.cs.pub.ro/courses/pm/prj2026/farhad\\_ali.gul/alexandru.ionita04](http://ocw.cs.pub.ro/courses/pm/prj2026/farhad_ali.gul/alexandru.ionita04) 

Last update: **2026/05/18 14:57**