

Proiect EEA  
Piano circuit  
Universitatea Politehnică București  
Facultatea de Automatică și Calculatoare

Duțu Alin Călin — Iftimie Adelin Cosmin

5 ianuarie 2021

***Abstract.*** Prin electronică se pot obține sunete folosind un număr mic de piese necesare. În acest proiect am realizat construirea a două circuite, ambele reprezentând un pian, folosind Tinkercad și o serie de piese prezentate în construcția fiecărui circuit.

***Cuvinte cheie:*** Piano, Temporizator 555, Arduino

# Cuprins

<b>1</b>	<b>Pian cu temporizator 555</b>	<b>2</b>
1.1	Descrierea componentelor . . . . .	2
1.2	Descrierea circuitului . . . . .	3
1.3	Link-uri și scheme . . . . .	3
<b>2</b>	<b>Pian cu Arduino</b>	<b>4</b>
2.1	Descrierea componentelor . . . . .	4
2.2	Descrierea circuitului . . . . .	4
2.3	Link-uri și scheme . . . . .	7
	<b>Mulțumiri</b>	<b>7</b>
	<b>Bibliografie</b>	<b>8</b>

# Capitolul 1

## Pian cu temporizator 555

### 1.1 Descrierea componentelor

Schema circuitului cuprinde:

1. 12 butoane
2. 9 rezistoare de  $1K\Omega$
3. 3 rezistoare de  $2K\Omega$
4. 1 rezistor de  $6,2K\Omega$
5. 1 condensator de  $100\text{ nF}$
6. 1 baterie de 9V si  $25\Omega$
7. 1 temporizator 555
8. 1 difuzor
9. 2 plăci Breadboard
10. 52 de fire de dimensiuni diferite

**Temporizatorul 555** reprezintă un cip folosit cu scopul de a crea impulsuri de durate diferite pentru a crea o formă de undă continuă între stările joase sau înalte. Temporizatorul 555 este foarte flexibil, ieftin și ușor de găsit, ceea ce îl face o piesă de circuit cu un punct de plecare favorabil pentru proiecte mici precum : ceasuri sau proiecte audio. Atașând dispozitivului rezistențe și condensatori în diferite moduri se pot obține 3 moduri diferite de funcționare:

1. **Modul Monostabil** - folosit pentru a crea întârzieri. În acest mod, un declanșator extern face ca cipul să emită un impuls de o durată reglabilă
2. **Modul Astable** - emite un semnal sub formă de undă, cipul comută între stările înalte și joase la o frecvență reglabilă
3. **Modul Bistabil** - comută între stările înalte și joase, dar pentru două intrări

## 1.2 Descrierea circuitului

Schema pianului folosește temporizatorul 555 în modul Astable, deoarece fiecare notă are o frecvență principală, drept pentru care, prin manipularea frecvenței se pot obține sunete diferite. Frecvența produsă de temporizator se bazează pe valorile condensatorului și a celor două rezistențe R1 și R2, unde R1 este rezistența dintre pin-ul **Discharge** al temporizatorului și borna pozitivă a bateriei, iar R2 este rezistența echivalentă a rezistoarelor care sunt parcurse de la pin-ul **Discharge** al temporizatorului și butonul care este apăsat. În cazul în care sunt apăstate mai multe butoane deodată, curentul va parcurge circuitul până la butonul care are rezistența echivalentă R2 mai mică.

Formula care determină frecvența de ieșire a circuitului este:

$$frecventa = \frac{1}{7(R1 + R2)C}$$

## 1.3 Link-uri și scheme

Link-ul circuitului: <https://www.tinkercad.com/things/kGCP1YdP2NQ>

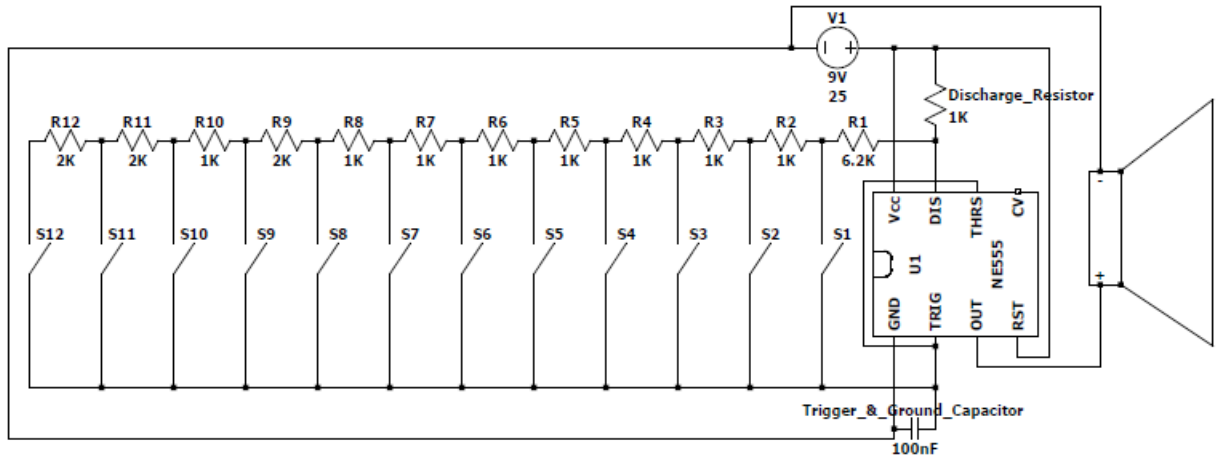


Figura 1.1: Schema circuitului implementată în LTSpice

## Capitolul 2

# Pian cu Arduino

### 2.1 Descrierea componentelor

Schema circuitului cuprinde:

1. 11 butoane
2. 11 LED-uri
3. 11 rezistoare de  $5K\Omega$
4. 1 rezistor de  $220\Omega$
5. 1 difuzor
6. 2 tăblițe Breadboard
7. 1 placa Arduino Uno R3
8. 53 de fire de dimensiuni diferite

**Arduino Uno** este un microcontroller provenit de la microcontroller-ul ATmega328P. Acesta are 14 intrări/ieșiri digitale sub formă de pini (din care 6 pot fi folosite ca ieșiri PWM), 6 intrări analogice, un rezonator ceramic de 16 MHz (CSTCE16M0V53-R0), un port USB, un port jack de alimentare, un port ICSP și un buton de resetare.

Această plăcuță vine la pachet și cu un program software cu un limbaj asemanator cu C/C++ prin care se poate manipula cu ușurință functionarea microcontroller-ului. De asemenea, există și o variantă cu limbaj Blockly care este cel mai ușor și mai intuitiv limbaj de programare pentru plăcuțele Arduino.

### 2.2 Descrierea circuitului

Acest circuit folosește Plăcuța Arduino pentru a scoate sunetele unui pian folosind un difuzor. Plăcuța a fost programată să scoată prin difuzor anumite frecvențe care sunt percepute de oameni ca niște sunete cu intensități diferite. Aceste frecvențe sunt eliberate în momentul în care se apasă butoanele circuitului.

Pentru acest circuit butonul cel mai din stânga scoate frecvența cea mai mică, iar butonul cel mai din dreapta scoate frecvența cea mai mare. Creșterea frecvențelor se face progresiv de la stânga la dreapta. În cazul în care se apasă mai multe butoane deodată se va "alege" butonul cu frecvența cea mai mică, deoarece programul din microcontroller verifică butoanele de la stânga la dreapta.

Programul folosit pentru setarea plăcuței Arduino este:

```
//Declararea de variabile
int button1 = 2; // Asocierea butoanelor cu numerele de pe
int button2 = 3; // placuta Arduino.
int button3 = 4;
int button4 = 5;
int button5 = 6;
int button6 = 7;
int button7 = 8;
int button8 = 9;
int button9 = 10;
int button10 = 11;
int button11 = 12;

int piezo = 13; // Asocierea difuzorului

int a, b, c, d, e, f, g, h, i, j, k; // Asocierea variabilelor pentru output

// Configurarea placii Arduino
void setup()
{
  // Initializarea comunicarii
  Serial.begin(9600);

  // Configurarea pinilor pentru receptionarea semnalului cu ajutorul butoanelor
  pinMode(button1, INPUT);
  pinMode(button2, INPUT);
  pinMode(button3, INPUT);
  pinMode(button4, INPUT);
  pinMode(button5, INPUT);
  pinMode(button6, INPUT);
  pinMode(button7, INPUT);
  pinMode(button8, INPUT);
  pinMode(button9, INPUT);
  pinMode(button10, INPUT);
  pinMode(button11, INPUT);
}

// Programarea placii Arduino
```

```

void loop()
{
// Citirea datelor de intrare prin variabilele de output
  a = digitalRead(button1);
  b = digitalRead(button2);
  c = digitalRead(button3);
  d = digitalRead(button4);
  e = digitalRead(button5);
  f = digitalRead(button6);
  g = digitalRead(button7);
  h = digitalRead(button8);
  i = digitalRead(button9);
  j = digitalRead(button10);
  k = digitalRead(button11);

// si trimiterea rezultatului catre Piezo a unei frecvente de vibrare
// astfel incat sa scoata sunete cat mai intense cu apasarea
// butoanelor de la stanga la dreapta sau nici un sunet daca nu e apasat
// nici un buton
  if(a == 1){
    tone(piezo,523);
  } else if(b == 1){
    tone(piezo,575);
  } else if(c == 1){
    tone(piezo,628);
  } else if(d == 1){
    tone(piezo,681);
  } else if(e == 1){
    tone(piezo,733);
  } else if(f == 1){
    tone(piezo,786);
  } else if(g == 1){
    tone(piezo,838);
  } else if(h == 1){
    tone(piezo,890);
  } else if(i == 1){
    tone(piezo,943);
  } else if(j == 1){
    tone(piezo,995);
  } else if(k == 1){
    tone(piezo,1047);
  } else{
    noTone(piezo);
  }
  delay(50);}

```

## 2.3 Link-uri și scheme

Link-ul circuitului: <https://www.tinkercad.com/things/dIhJH05zwqa-arduino-circuit>

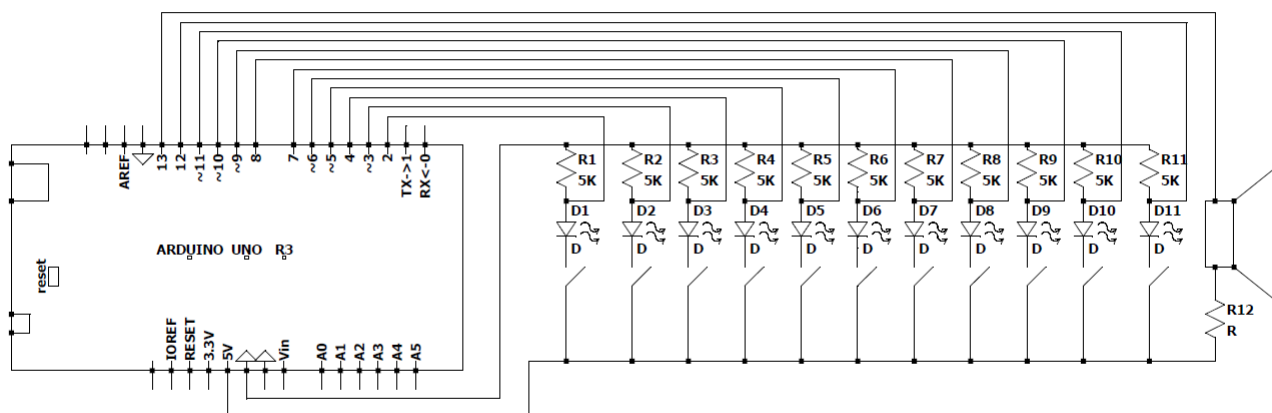


Figura 2.1: Schema circuitului implementată în LTSpice

## Mulțumiri

Am dori să adresăm mulțumiri domnului Isuf Vlad pentru îndrumarea și toate indicațiile oferite pentru realizarea acestui proiect.



## Bibliografie

- [1] Piano. <https://www.tinkercad.com/things/hPWb1diqMaK>.
- [2] Simple 555 piano. <https://www.tinkercad.com/things/i0Fygjxe6Pi>.
- [3] Joshua Brooks. Simple Electronic Piano. <https://www.instructables.com/Simple-Electronic-Piano/>.
- [4] Gabriela Ciuprina. Template Latex v5 (Ultima accesare: 05.01.2021). <https://acs.curs.pub.ro/2019/mod/resource/view.php?id=19677>.