

***Raport***  
***Sisteme încorporate***

Ciobanu Victor

Ciobanu Nicolae

Roga Andrei

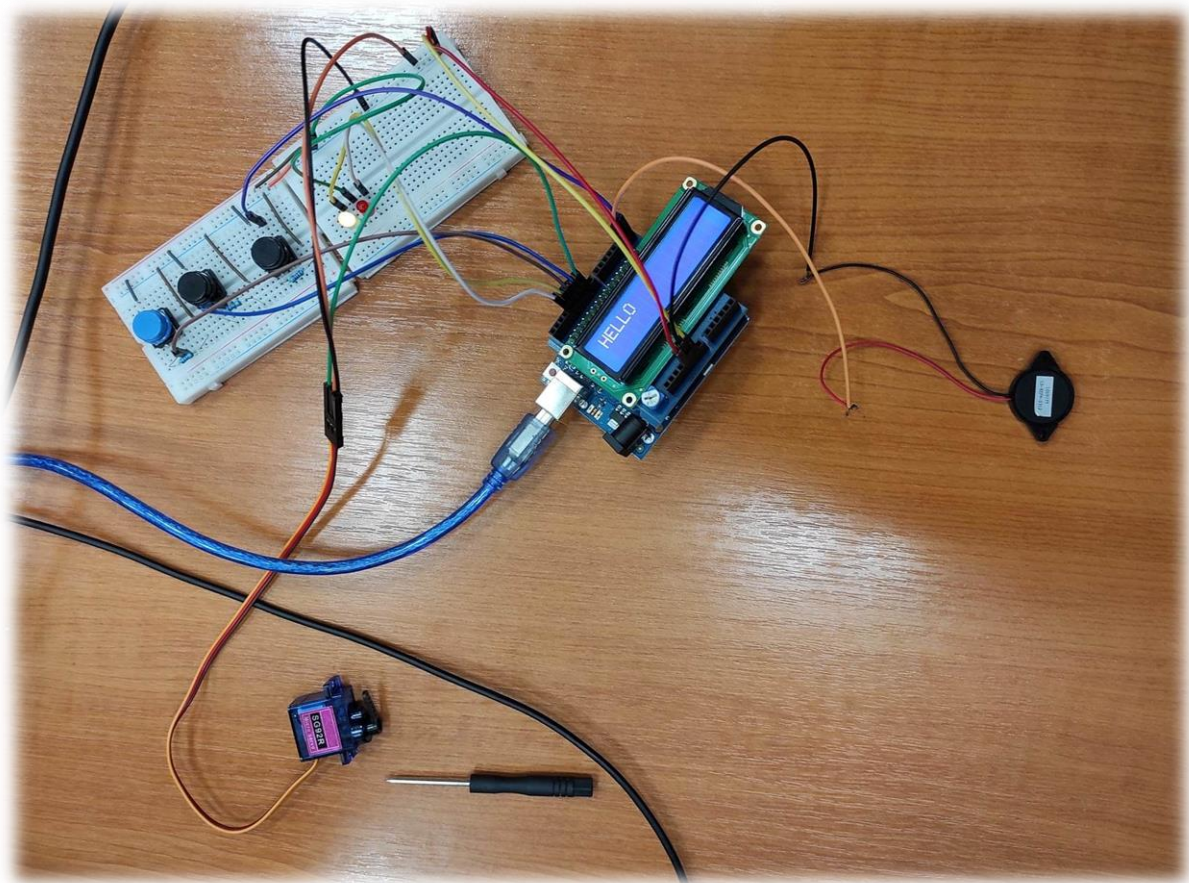
## Cerințe exercițiului

Când se alimentează sistemul se va aprinde LED-ul roșu.

Când se apasă pe primul buton sistemul va face următoarele:

- Se va aprinde LED-ul verde
- Pe LCD se va afișa un mesaj de pornire
- Când se va apăsa pe al doilea buton servomotorul se va deplasa spre stânga, iar difuzorul va cânta un anumit cântec
- Când se va apăsa pe al treilea buton servomotorul se va deplasa spre dreapta, iar difuzorul va cânta un alt cântec

## O figură realizată cu simulatorul Arduino



## Codul sursă

```
// Se includ librăriile pentru LCD, servomotor
#include <LiquidCrystal.h>
#include <Servo.h>
// Se include librăria cu note muzicale
#include "pitches.h"

// Se declară variabilele pentru pinii care vor furniza semnale
const int led_rosu = 13;
const int led_verde = 12;
const int buton1 = 0;
const int buton2 = 10;
const int buton3 = 11;
const int buzzer = 1;

// Se declară o variabilă pentru unghiul axului a servomotorului
int pos = 0;
// Se declară o variabilă pentru a memora starea proiectului on sau off
int onoff = 0;
// Se declară o variabilă pentru a memora starea de LOW a butonului
int change = 0;

// Se declară o variabilă de tip Servo pentru servomotor
Servo servo_9;
// Se declară o variabilă de tip LiquidCrystal cu pinii care vor fi utilizați
// pentru LCD
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

// Notele pentru prima melodie
int melodie1[] = {
    REST, NOTE_D4,
    NOTE_G4, NOTE_AS4, NOTE_A4,
    NOTE_G4, NOTE_D5,
    NOTE_C5,
    NOTE_A4,
    NOTE_G4, NOTE_AS4, NOTE_A4,
    NOTE_F4, NOTE_GS4,
    NOTE_D4,
    NOTE_D4,

    NOTE_G4, NOTE_AS4, NOTE_A4,
    NOTE_G4, NOTE_D5,
    NOTE_F5, NOTE_E5,
    NOTE_DS5, NOTE_B4,
    NOTE_DS5, NOTE_D5, NOTE_CS5,
    NOTE_CS4, NOTE_B4,
    NOTE_G4,
```

```

NOTE_AS4,

NOTE_D5, NOTE_AS4,
NOTE_D5, NOTE_AS4,
NOTE_DS5, NOTE_D5,
NOTE_CS5, NOTE_A4,
NOTE_AS4, NOTE_D5, NOTE_CS5,
NOTE_CS4, NOTE_D4,
NOTE_D5,
REST, NOTE_AS4,

NOTE_D5, NOTE_AS4,
NOTE_D5, NOTE_AS4,
NOTE_F5, NOTE_E5,
NOTE_DS5, NOTE_B4,
NOTE_DS5, NOTE_D5, NOTE_CS5,
NOTE_CS4, NOTE_AS4,
NOTE_G4
};
//Durata notelor pentru prima melodie
int durate1[] = {
    2, 4,
    4, 8, 4,
    2, 4,
    2,
    2,
    4, 8, 4,
    2, 4,
    1,
    4,

    4, 8, 4,
    2, 4,
    2, 4,
    2, 4,
    4, 8, 4,
    2, 4,
    1,
    4,

    2, 4,
    2, 4,
    2, 4,
    2, 4,
    4, 8, 4,
    2, 4,
    1,
    4, 4,

```

```

2, 4,
2, 4,
2, 4,
2, 4,
4, 8, 4,
2, 4,
1
};
// Notele pentru a doua melodie
int melodie2[] = {
    NOTE_G4, NOTE_C4, NOTE_DS4, NOTE_F4, NOTE_G4, NOTE_C4, NOTE_DS4, NOTE_F4,
    NOTE_G4, NOTE_C4, NOTE_DS4, NOTE_F4, NOTE_G4, NOTE_C4, NOTE_DS4, NOTE_F4,
    NOTE_G4, NOTE_C4, NOTE_E4, NOTE_F4, NOTE_G4, NOTE_C4, NOTE_E4, NOTE_F4,
    NOTE_G4, NOTE_C4, NOTE_E4, NOTE_F4, NOTE_G4, NOTE_C4, NOTE_E4, NOTE_F4,
    NOTE_G4, NOTE_C4,

    NOTE_DS4, NOTE_F4, NOTE_G4, NOTE_C4, NOTE_DS4, NOTE_F4,
    NOTE_D4,
    NOTE_F4, NOTE_AS3,
    NOTE_DS4, NOTE_D4, NOTE_F4, NOTE_AS3,
    NOTE_DS4, NOTE_D4, NOTE_C4,

    NOTE_G4, NOTE_C4,

    NOTE_DS4, NOTE_F4, NOTE_G4, NOTE_C4, NOTE_DS4, NOTE_F4,
    NOTE_D4,
    NOTE_F4, NOTE_AS3,
    NOTE_DS4, NOTE_D4, NOTE_F4, NOTE_AS3,
    NOTE_DS4, NOTE_D4, NOTE_C4,
    NOTE_G4, NOTE_C4,
    NOTE_DS4, NOTE_F4, NOTE_G4, NOTE_C4, NOTE_DS4, NOTE_F4,

    NOTE_D4,
    NOTE_F4, NOTE_AS3,
    NOTE_D4, NOTE_DS4, NOTE_D4, NOTE_AS3,
    NOTE_C4,
    NOTE_C5,
    NOTE_AS4,
    NOTE_C4,
    NOTE_G4,
    NOTE_DS4,
    NOTE_DS4, NOTE_F4,
    NOTE_G4,

    NOTE_C5,
    NOTE_AS4,
    NOTE_C4,
    NOTE_G4,
    NOTE_DS4,

```

```

NOTE_DS4, NOTE_D4,
NOTE_C5, NOTE_G4, NOTE_GS4, NOTE_AS4, NOTE_C5, NOTE_G4, NOTE_GS4, NOTE_AS4,
NOTE_C5, NOTE_G4, NOTE_GS4, NOTE_AS4, NOTE_C5, NOTE_G4, NOTE_GS4, NOTE_AS4,

REST, NOTE_GS5, NOTE_AS5, NOTE_C6, NOTE_G5, NOTE_GS5, NOTE_AS5,
NOTE_C6, NOTE_G5, NOTE_GS5, NOTE_AS5, NOTE_C6, NOTE_G5, NOTE_GS5, NOTE_AS5
};
// Durata notelor pentru a doua melodie
int durate2[] = {
    8, 8, 16, 16, 8, 8, 16, 16,
    8, 8, 16, 16, 8, 8, 16, 16,
    8, 8, 16, 16, 8, 8, 16, 16,
    8, 8, 16, 16, 8, 8, 16, 16,
    4, 4,

    16, 16, 4, 4, 16, 16,
    1,
    4, 4,
    16, 16, 4, 4,
    16, 16, 1,

    4, 4,

    16, 16, 4, 4, 16, 16,
    1,
    4, 4,
    16, 16, 4, 4,
    16, 16, 1,
    4, 4,
    16, 16, 4, 4, 16, 16,

    2,
    4, 4,
    8, 8, 8, 8,
    1,
    2,
    2,
    2,
    2,
    2,
    2,
    4, 4,
    1,

    2,
    2,
    2,
    2,
    2,
    2,
    4, 4,

```

```

8, 8, 16, 16, 8, 8, 16, 16,
8, 8, 16, 16, 8, 8, 16, 16,

4, 16, 16, 8, 8, 16, 16,
8, 16, 16, 16, 8, 8, 16, 16
};

void setup() {
    // Pinii de led se declară ca fiind de tip OUTPUT
    pinMode(led_rosu, OUTPUT);
    pinMode(led_verde, OUTPUT);
    // Pinii butoanelor se declară ca fiind de tip INPUT
    pinMode(buton1, INPUT);
    pinMode(buton2, INPUT);
    pinMode(buton3, INPUT);
    // Se declară pinul servomotorului ca fiind 9
    servo_9.attach(9, 500, 2500);
    // Se declară câte caractere și câte linii afișează LCD
    lcd.begin(16, 2);
    // Pinul difuzorului se declară ca fiind de tip OUTPUT
    pinMode(buzzer, OUTPUT);
}

void loop() {
    // Dacă se apasă butonul 1 și starea anterioară a fost LOW atunci se execută
    comenziile
    if (digitalRead(buton1) == HIGH && change == 1) {
        // Starea anterioară se schimbă pe 0
        change = 0;
        // Se schimbă starea proiectului on sau off
        onoff = !onoff;
        // Se șterge tot de pe LCD
        lcd.clear();
        // Se afișează "HELLO" pe LCD
        lcd.print("HELLO");
    }
    // Dacă starea anterioară este LOW, atunci se schimbă starea anterioară pe 1
    if (digitalRead(buton1) == LOW) {
        change = 1;
    }
    // Dacă starea proiectului este pe on, atunci se execută următoarele
    comenzi
    if (onoff == 1)
    {
        // Se stinge ledul rosu
        digitalWrite(led_rosu, LOW);
        // Se aprinde ledul verde
        digitalWrite(led_verde, HIGH);
    }
}

```

```

// Dacă se apasă butonul 2, atunci se execută următoarele comenzi
if (digitalRead(buton2) == HIGH)
{
    // Se mișcă axul servomotorului într-o direcție cu 90 de grade
    for (pos = 90; pos < 180; pos += 1)
    {
        servo_9.write(pos);
        delay(15);
    }
    // Se difuzează melodia 1 pe difuzor
    int size = sizeof(durate1) / sizeof(int);
    for (int note = 0; note < size; note++)
    {
        int durata = 1000 / durate1[note];
        tone(buzzer, melodie1[note], durata);
        int pauzaDintreNote = durata * 1.30;
        delay(pauzaDintreNote);
        noTone(buzzer);
    }
}

// Dacă se apasă butonul 3, atunci se execută următoarele comenzi
if (digitalRead(buton3) == HIGH)
{
    // Se mișcă axul servomotorului într-o cealaltă direcție cu 90 de grade
    for(pos = 90; pos >= 0; pos -= 1)
    {
        servo_9.write(pos);
        delay(15);
    }
    // Se difuzează melodia 2 pe difuzor
    int size = sizeof(durate2) / sizeof(int);
    for (int note = 0; note < size; note++)
    {
        int durata = 1000 / durate2[note];
        tone(buzzer, melodie2[note], durata);
        int pauzaDintreNote = durata * 1.30;
        delay(pauzaDintreNote);
        noTone(buzzer);
    }
}

// Dacă starea proiectului este pe off, atunci se execută următoarele
comenzi
else
{
    // Se șterge tot de pe LCD
    lcd.clear();
    // Se aprinde ledul roșu
    digitalWrite(led_rosu, HIGH);
}

```

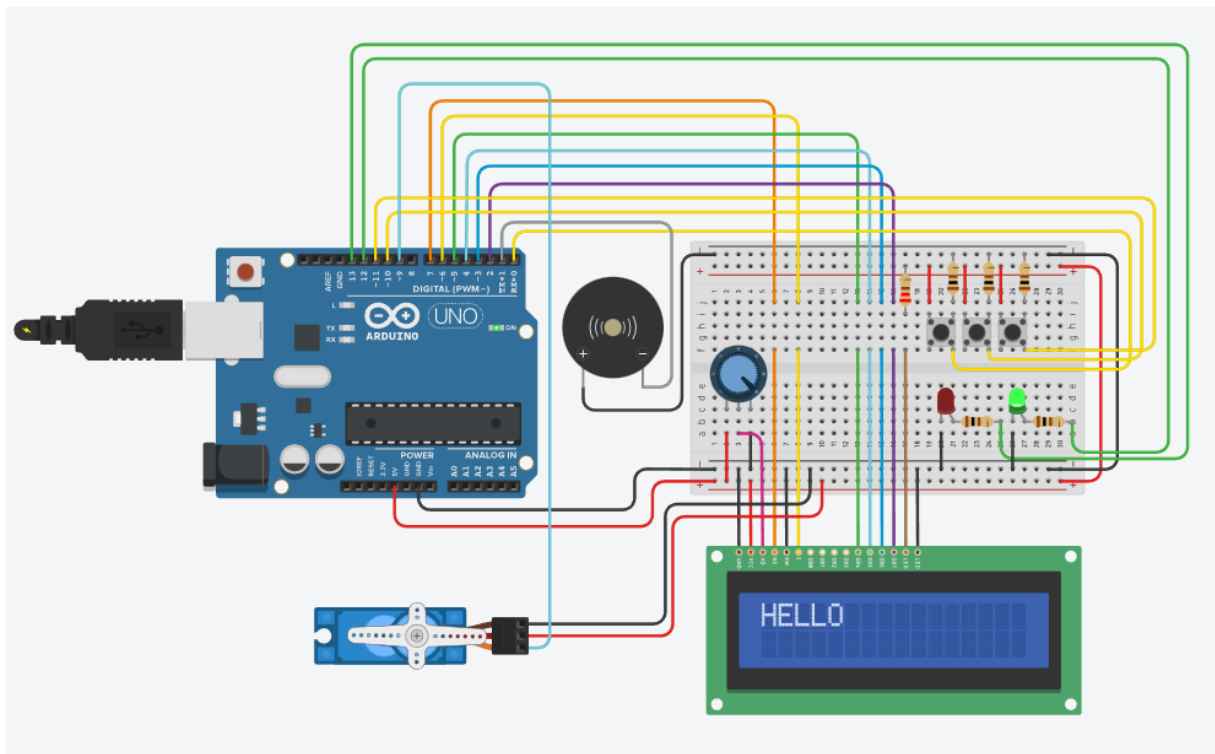


```

// Se stinge ledul verde
digitalWrite(led_verde, LOW);
}
}

```

## Link-ul cu sistemul de pe Tinkercad



<https://www.tinkercad.com/things/5P9kz64O49R?sharecode=bqYY8MJuJshhEvLtkRPNNQ2tvLbwbWQ1sbkQu4sp6Vo>