

Tuneduino

Introducere

Tuneduino este un music player ce utilizează fișiere de tipul `.wav`, proiectat pentru a oferi o experiență simplă de ascultare a muzicii. Scopul acestuia este de a oferi o soluție de redare a fișierelor audio în format WAV, utilizând un Arduino. Ideea a pornit de la dorința de a crea un sistem de redare a muzicii care să fie ușor de folosit, economic și personalizabil.

Tuneduino este util atât pentru utilizatori obișnuiți, cât și pentru pasionații de electronice și programare, întrucât reprezintă un dispozitiv simplu și personalizabil. Fiind open-source, poate fi personalizat după bunul plac, cu adăugare de noi funcționalități.

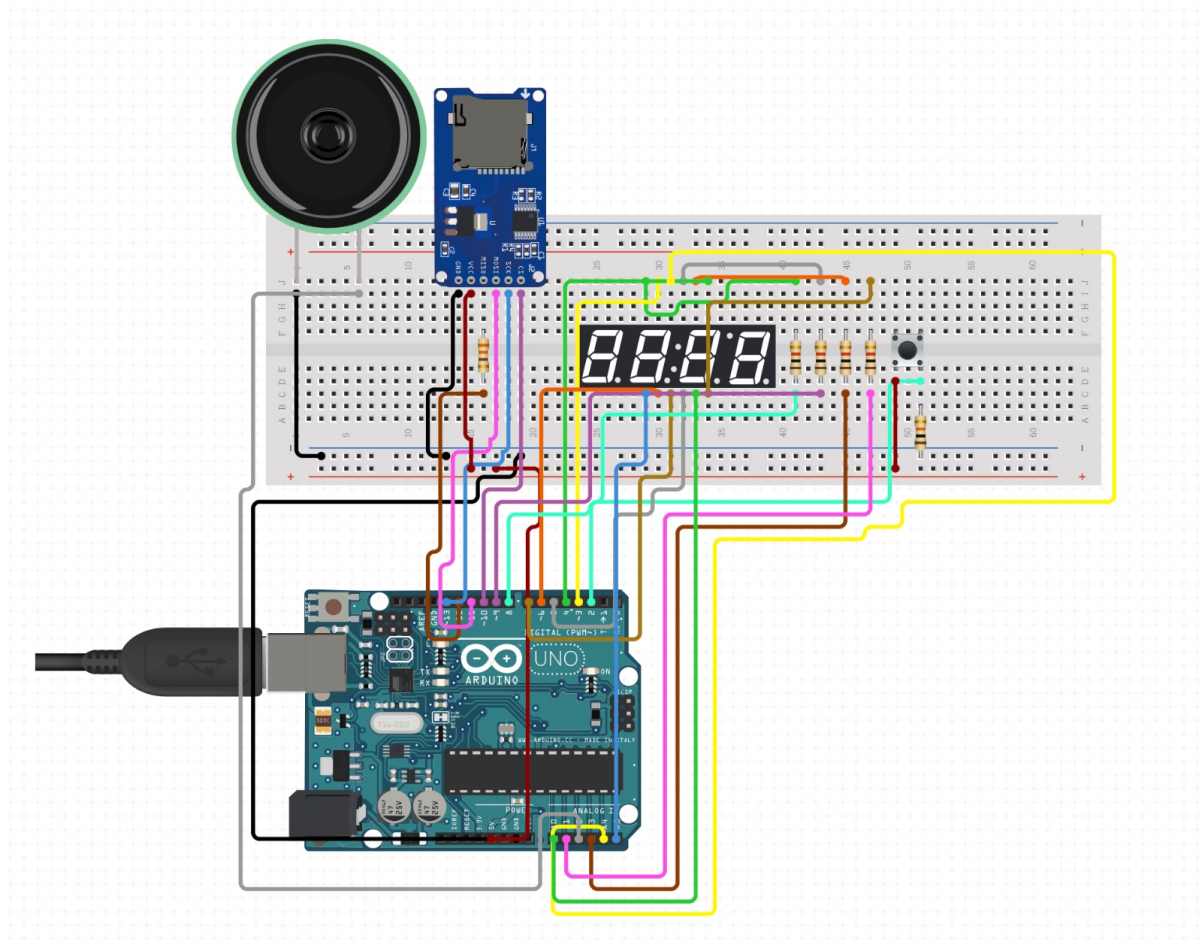
Descriere generală



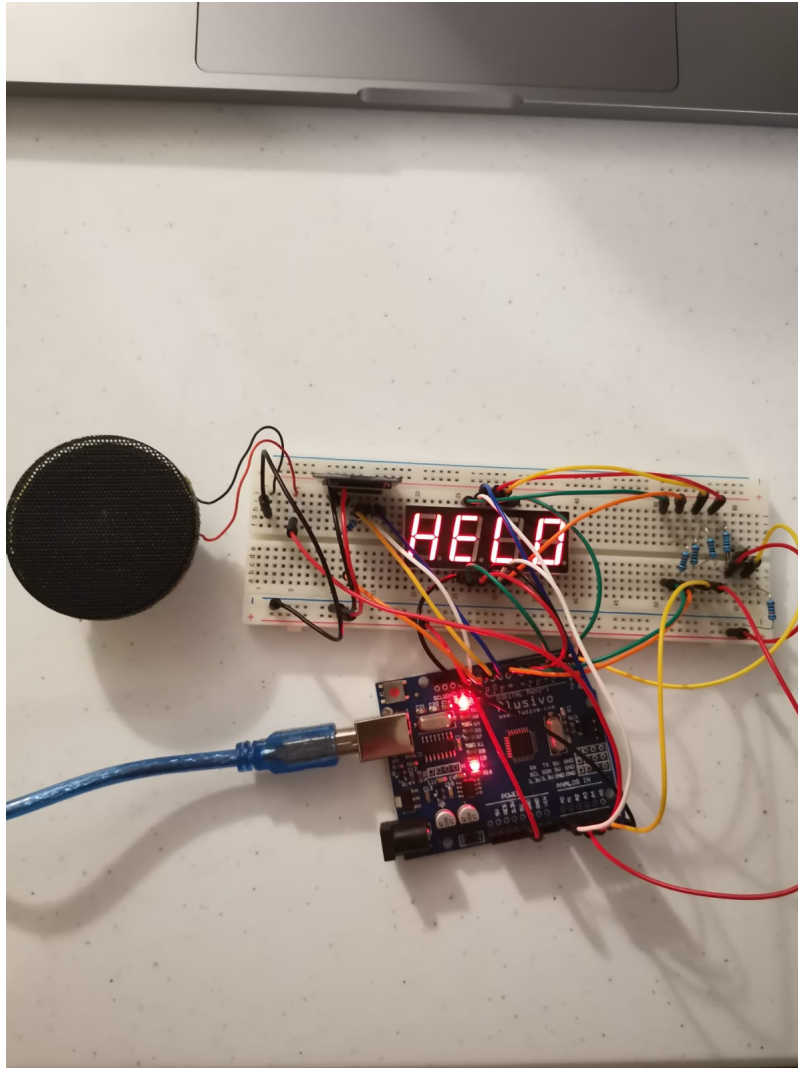
Hardware Design

- 1 x Arduino Uno
- 4 x butoane
- 1 x modul pentru SD card
- 1 x SD card
- 1 x breadboard
- 1 x speaker

Schema electrica



Montaj



Software Design

- Arduino IDE
- DFRobotDFPlayerMini (pentru player-ul de muzica), SevSeg (pentru operarea cu display-ul pe 7 segmente)

```
void loop()
{
  unsigned long currentMillis = millis();
  unsigned long duration;
  if (in_menu) {
    buttonState = digitalRead(mainbutton);
```

```
    if (buttonState != lastButtonState) { // button state changed
      updateState();
    }
```

```
    if (startPressed == -1 && endPressed == -1) {
      if (longPress) {
        in_menu = false;
        playing = true;
```

```
    myDFPlayer.play(song_number); // Play song
    if (myDFPlayer.available()) {
        printDetail(myDFPlayer.readType(), myDFPlayer.read()); //Print the
detail message from DFPlayer to handle different errors and states.
    }
} else {
    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;
        song_number++;
        if (song_number > 9) {
            song_number = 1;
        }
    }
}
```

```
    sevseg.setNumber(song_number);
}
startPressed = 0;
endPressed = 0;
}
```

```
lastButtonState = buttonState;
} else if (playing) {
    sevseg.blank();
    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;
        safety_check = true;
    }
}
```

```
if (digitalRead(mainbutton) == LOW && safety_check) {
    playing = false;
    myDFPlayer.stop(); // Stop playing
    safety_check = false;
}
} else if (!playing && !in_menu) {
    sevseg.setChars("P");
}
```

```
if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    safety_check = true;
}
```

```
if (digitalRead(mainbutton) == LOW && safety_check) {
    playing = true;
    // myDFPlayer.play(song_number); // Resume playing
    myDFPlayer.next();
    if (myDFPlayer.available()) {
        printDetail(myDFPlayer.readType(), myDFPlayer.read()); //Print the
detail message from DFPlayer to handle different errors and states.
    }
    sevseg.blank();
}
```

```
    safety_check = false;
  }
}
```

```
sevseg.refreshDisplay();
}
```

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună 😊.

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

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/prj2024/vstoica/adelin_costin.popa



Last update: **2024/05/23 21:43**