

Cutia de bijuterii muzicală

Introducere

- Cutia de bijuterii muzicală este concepută pentru a ne ușura viața și a ne crea o experiență plăcută la utilizare. La simpla apropiere de cutie aceasta se va deschide automat pentru a putea depozita acolo bijuteriile. De asemenea, la fiecare deschidere va începe o melodie simulând astfel cutiuța muzicală.
- Ideea a venit o dată cu laboratoarele [Laborator 3](#) , [Laborator 2](#)
- Cutia de bijuterii muzicală este utilă pentru depozitarea și sortarea cât mai rapidă a bijuteriilor. Totodată aceasta te introduce într-o lume de basm cu ajutorul sunetului.

Descriere generală

Printr-un senzor de mișcare se va detecta prezența unei persoane în apropierea cutiei și se va porni automat un servomotor conectat printr-un fir de capacul cutiei. Odată deschisă cutia va începe o melodie preluată din repo-ul public [Github](#)

Schema bloc:



Hardware Design

Piese folosite:

1. Arduino UNO
2. Senzor mișcare
3. Servomotor
4. Difuzor
5. Cabluri

Schemă circuit:



Schema electrică:



Software Design

Descrierea codului aplicației (firmware):

- Mediu de dezvoltare: Arduino 1.8.14
- Bibliotecă folosită: Servo Library
- Se realizează acționarea motorului care depinde de distanța utilizatorului.

Idee principale ale dezvoltării cu ajutorul funcțiilor:

void setup()

Setare parametrilor de I/O

```
sg90.attach(8);  
pinMode(trig, OUTPUT);  
pinMode(echo, INPUT);
```

void loop()

Se calculează distanța și durata interceptării senzorului ultrasonic

```
duration = pulseIn(echo, 1);  
distance = duration * 0.034 / 2;
```

Parametrii de acționare

```
if (distance < 30)
```

Explicație a melodiei, calculului duratei plus funcțiile folosite:

Sizeof oferă numărul de octeți, fiecare valoare **int** este compusă din doi octeți (16 biți). Există două valori pe notă (**înălțimea și durata**), deci pentru fiecare notă sunt patru octeți.

```
int notes = sizeof(melody) / sizeof(melody[0]) / 2;
```

Calculează durata unei **note întregi** în ms

```
int wholenote = ( 60000 * 4 ) / tempo;
```

Selectare notă, prelucrare și emiterie finală

```
// Iterează până la finalul notelor melodiei
// Vectorul de note conține atât notele cât și durata fiecăreia
for (int thisNote = 0; thisNote < notes * 2; thisNote = thisNote + 2)
{
    // Calculăm durata fiecăreii note
    divider = pgm_read_word_near(melody+thisNote + 1);
    if (divider > 0) {
        // Notă obișnuită, merge mai departe
        noteDuration = (wholenote) / divider;
        // Dacă nota este cu durată negativă atunci se ia valoarea pozitivă
        // însă se mărește durata cu jumătate
    } else if (divider < 0) {
        noteDuration = (wholenote) / abs(divider);
        noteDuration *= 1.5;
    }
    // Cântăm nota doar pentru 90% din durată, lăsând 10% ca pauză
    tone(buzzer, pgm_read_word_near(melody+thisNote), noteDuration * 0.9);

    // Așteptăm durata specificată înainte de a reda următoarea notă
    delay(noteDuration);

    // Oprește generarea formei de undă înainte de următoarea notă
    noTone(buzzer);
}
```

Pentru a accesa mai rapid variabila stocată în SRAM dar și pentru a susține cantitatea mare de date ce conține notele muzicale nemodificate dinamic în timpul execuției am folosit funcția:

```
pgm_read_word_near()
```

Am generat o undă pătrată cu frecvența de **pgm_read_word_near(melody + thisNote)** pentru pinul *buzzer* și o durată de 90%, lăsând 10% ca pauză.

```
tone(buzzer, pgm_read_word_near(melody+thisNote), noteDuration * 0.9);
```

Dacă nu specific durata, tonul va continua până când funcția **noTone** este apelată pe același pin *buzzer*.

```
noTone(buzzer);
```

Rezultate Obținute

Rezultatele se pot urmări aici: [YouTube](#)

Concluzii

Proiectul s-a dovedit a fi o provocare din care am acumulat cunoștințe noi de lucru cu Arduino.

Download

Proiectul poate fi descărcat de aici: [Arhiva](#)

Jurnal

Creare: 28.04.2022

Modificări: 16.05.2022

Bibliografie/Resurse

https://ocw.cs.pub.ro/courses/pm/prj2022/robert/jewelry_box

<https://ocw.cs.pub.ro/courses/pm/lab/lab2-2022> <https://ocw.cs.pub.ro/courses/pm/lab/lab3-2022>

<https://github.com/robsoncouth/arduino-songs/tree/master/furelise>

[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2022/robert/jewelry_box



Last update: **2022/05/20 17:39**