

Jucarie pentru bebelusi

Nastase Ana-Maria
333CC

Introducere

Dispozitivul creat de mine reprezinta o jucarie pentru linistirea si supravegherea bebelusilor, care detecteaza plansul copilului si incepe sa redea muzica, insotita de lumini colorate, pentru a capta atentia bebelusului, pana cand parintele ajunge in camera si poate opri jucaria apasand un buton. Dispozitivul va alerta parintele, printr-o notificare trimisa pe telefon prin bluetooth.

Ideea mi-a venit dupa ce am vazut o jucarie asemanatoare, care se ataseaza de patutul bebelusului si incepe sa cante atunci cand acesta plange. Eu am decis sa extind aceasta idee, adaugand atat luminile, cat si functionalitatea de dispozitiv de monitorizare, care anunta parintele despre plansul copilului.

Cred ca o astfel de jucarie este un ajutor pentru orice parinte de bebelus, deoarece, inca din faza de nou-nascuti, copiii sunt fascinati de muzica si de lumini contrastante si pot fi astfel tinuti "ocupati" si linistiti pentru o perioada scurta de timp. Si faptul ca adultul primeste o notificare pe telefon cand micutul plange este foarte util, chiar daca jucaria nu ofera toate avantajele unui monitor pentru bebelusi care permite parintelui sa vada si sa asculte copilul in orice moment, de la distanta. Pe de alta parte, dispozitivul este folositor pentru un parinte care este acasa, dar nu aude bebelusul plangand pentru ca, spre exemplu, face dus sau gateste. In plus, o astfel de jucarie implica niste costuri reduse in comparatie cu un dispozitiv care inregistreaza audio-video.

Descriere generală

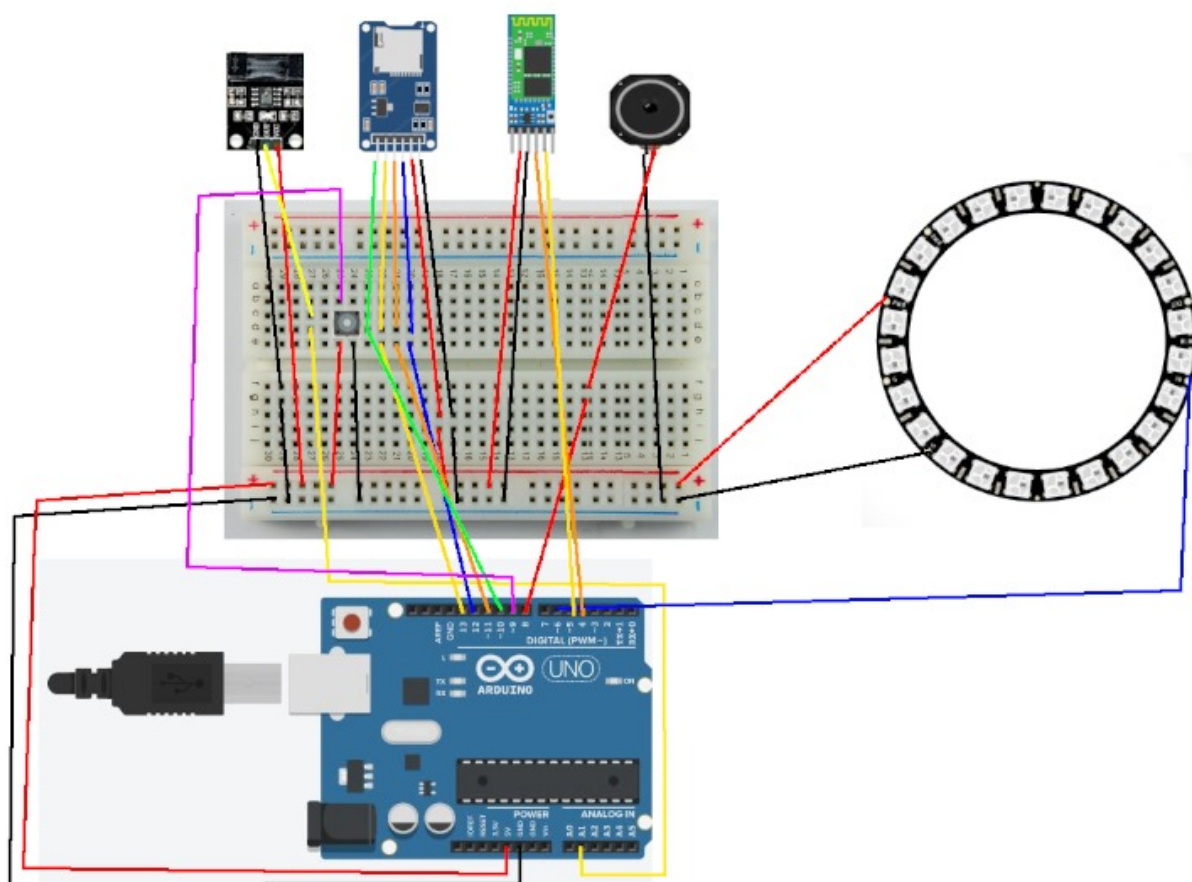


Arduino va primi de la senzorul de sunet semnalul analogic, pe care il va converti in digital cu ajutorul ADC-ului inclus. Pentru a determina daca bebelusul plange, se va verifica daca numarul citirilor senzorului mai mari decat o anumita valoare, pe o durata de 5 secunde, depaseste un numar minim. Daca da, atunci microcontroller-ul va face cercul de LED-uri sa lumineze si va incepe sa redea melodii de pe un card SD, cu ajutorul modului Micro SD (comunicand prin protocolul SPI) si al difuzorului. De asemenea, va trimite o alerta catre un telefon conectat prin Bluetooth (folosind un modul cu interfata UART), pentru a semnala plansul copilului. Dispozitivul poate fi oprit cu ajutorul unui buton, iar, daca se doreste ca acesta sa intre din nou in stare de functionare, de asteptare a plansului, trebuie apasat acelasi buton.

Hardware Design

Piese utilizate:

- Arduino Uno R3 ATMEGA328P
- Modul microfon, senzor sunet
- Modul Micro SD
- Card Micro SD 8GB
- Mini Difuzor Brick
- NeoPixel Ring - 24 x WS2812 5050 RGB LED Adresabile
- Modul Bluetooth BT 06



Software Design

Mediul de dezvoltare utilizat a fost Arduino IDE. In ceea ce priveste bibliotecile, acestea sunt:

- Adafruit_NeoPixel.h - biblioteca folosita pentru a controla inelul de led-uri Neopixel Ring
- SPI.h si SD.h - utilizate pentru comunicarea cu modulul SD
- TMRpcm.h - biblioteca necesara redarii de muzica de pe un card SD
- SoftwareSerial.h - pentru comunicarea cu modulul Bluetooth

Pentru a primi pe telefon un mesaj de avertizare atunci cand jucaria porneste, am instalat pe telefon o aplicatie numita "Arduino Bluetooth Controller", prin care pot fi vizualizate mesajele trimise prin interfata seriala.

Codul sursa:

[baby_toy.ino](#)

```
#include <Adafruit_NeoPixel.h>
#include <SPI.h>
#include <SD.h>
#include <TMRpcm.h>
#include <SoftwareSerial.h>

#define SENSOR A1 // senzorul de sunet
#define LED_PIN 6 // input-ul inelului de led-uri
#define BUTTON 2 // butonul on/off
#define SS 10
#define SD_ChipSelectPin 4
#define SPEAKER 9

#define LED_COUNT 24
#define LED_DELAY 10 // intervalul dupa care isi schimba culoarea
fiecare led
#define CRYING_WAIT_TIME 5000 // 5 secunde
#define CRYING_THRESHOLD 10 // nr de citiri > 200 in 5 secunde necesare
pt a porni jucaria
#define CRYING_INTENSITY 200 // intensitatea minima a sunetului
#define DEBOUNCE_TIME 500

SoftwareSerial mySerial(1, 0); // RX, TX pentru bluetooth
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800); //
inelul de led-uri
TMRpcm tmrpcm; // obiect folosit pt redarea muzicii
unsigned long ledInterval;
long firstPixelHue;
unsigned long soundCount; // nr de citiri > 200 in 5 secunde
unsigned long cryingTime;
volatile bool onOff;
bool crying;
volatile unsigned long lastInterrupt;

void setup() {
  pinMode(SENSOR, INPUT);
  pinMode(BUTTON, INPUT_PULLUP);
  pinMode(SS, OUTPUT);
  pinMode(SPEAKER, OUTPUT);

  onOff = 0;
  crying = 0;
```

```
strip.begin();
strip.show();
strip.setBrightness(10);

ledInterval = millis();
firstPixelHue = 0;

Serial.begin(9600);
mySerial.begin(9600);

soundCount = 0;
cryingTime = millis();

lastInterrupt = 0;
attachInterrupt(digitalPinToInterrupt(BUTTON), isr, RISING); //
intrerupere la apasarea butonului

if (!SD.begin(SD_ChipSelectPin)) {
    Serial.println("SD fail");
    return;
} else {
    Serial.println("SD ok");
}
tmrpcm.speakerPin = SPEAKER;
tmrpcm.quality(1);
tmrpcm.setVolume(6);
}

void loop() {
    if (onOff == 1) {
        // LED-urile
        if (millis() - ledInterval >= LED_DELAY) {
            ledInterval = millis();

            for(int i = 0; i < LED_COUNT; i++) {
                if (crying == 1) {
                    // Daca bebelusul plange, se aprind led-urile
                    int pixelHue = firstPixelHue + (i * 65536L / LED_COUNT);
                    strip.setPixelColor(i, strip.gamma32(strip.ColorHSV(pixelHue)));
                } else {
                    // primul led este verde pentru a semnala functionarea
                    dispozitivului
                    if (i == 0) {
                        strip.setPixelColor(0, 0, 255, 0);
                    } else {
                        // celelalte led-uri sunt stinse
                        strip.setPixelColor(i, 0, 0, 0);
                    }
                }
            }
        }
    }
}
```

```
strip.show();

firstPixelHue += 256;
if (firstPixelHue >= 5*65536) {
    firstPixelHue = 0;
}
}

// senzorul de sunet
if (crying == 0) {
    int soundLevel = analogRead(SENSOR);
    if (soundLevel >= CRYING_INTENSITY) {
        soundCount++;
    }

    // Se verifica daca au trecut cele 5 secunde
    if (millis() - cryingTime >= CRYING_WAIT_TIME) {
        if (soundCount >= CRYING_THRESHOLD) { // nr citirilor
senzorului
            // Jucaria porneste
            crying = 1;

            // se initializeaza variabilele necesare led-urilor
            ledInterval = millis();
            firstPixelHue = 0;

            // porneste muzica
            tmrpcm.play("song.wav", 15);

            // se trimite notificare
            Serial.println("Your baby is crying!");
        }

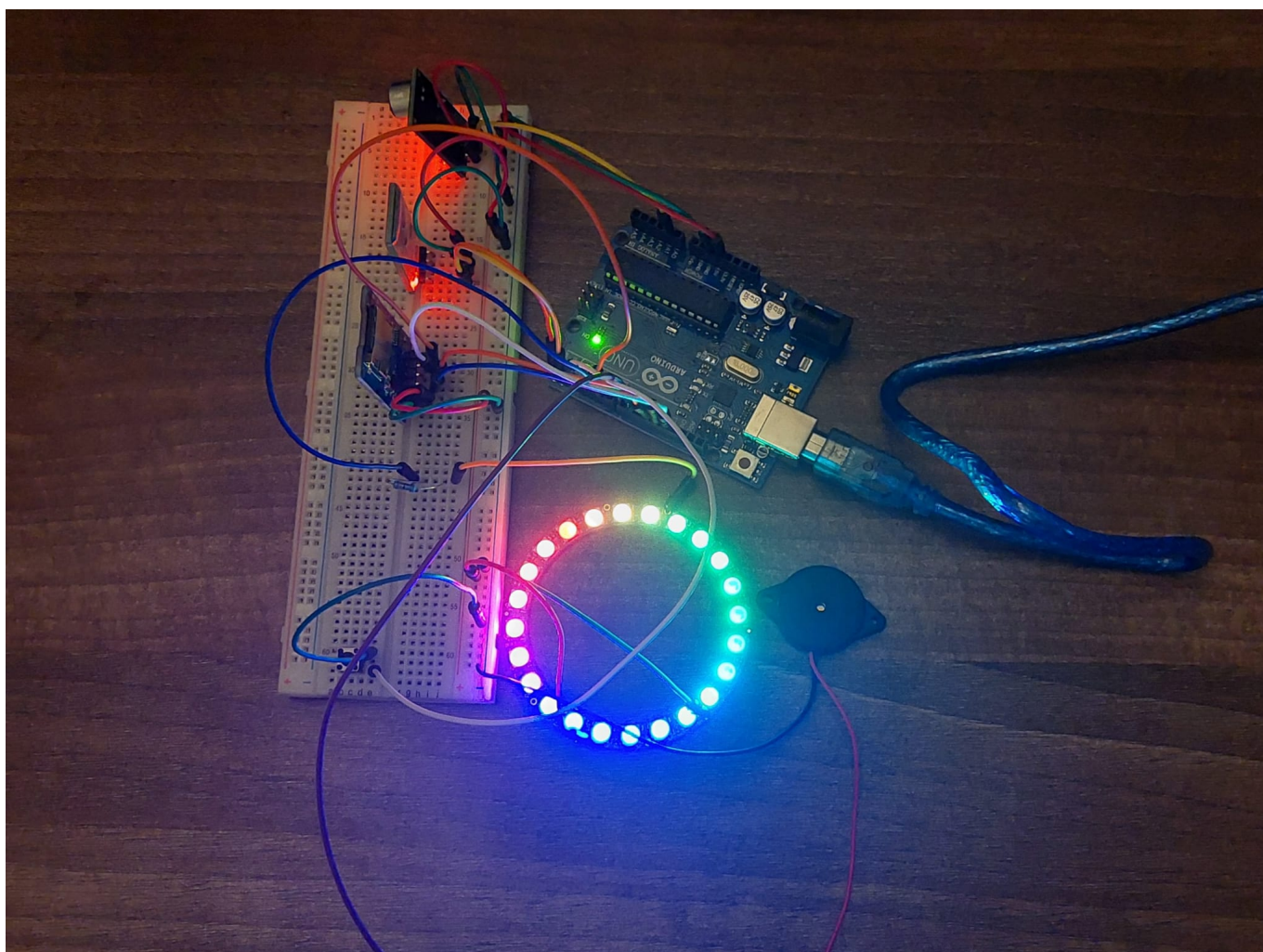
        cryingTime = millis();
        soundCount = 0;
    }
}
} else {
    crying = 0;
    ledInterval = millis();
    firstPixelHue = 0;
    soundCount = 0;
    cryingTime = millis();

    // se opreste muzica
    tmrpcm.stopPlayback();

    // led-urile sunt stinse
    for(int i = 0; i < LED_COUNT; i++)
        strip.setPixelColor(i, 0, 0, 0);
    strip.show();
}
```

```
}  
}  
  
void isr() {  
    // debounce pt buton  
    if (millis() - lastInterrupt >= DEBOUNCE_TIME) {  
        lastInterrupt = millis();  
        onOff = !onOff;  
    }  
}
```

Rezultate Obținute



Concluzii

Dispozitivul functioneaza conform descrierii, desi cu siguranta nu este cu adevarat o jucarie ce ar putea fi folosita intr-o situatie reala, atat prin prisma aspectului, cat si din cauza volumului redus al

sunetului redat (problema ce ar fi putut fi corectata utilizand un amplificator sau un difuzor mai performant). In ciuda acestor aspecte, un astfel de dispozitiv este fezabil si util, iar pentru mine realizarea proiectului a reprezentat o experienta foarte buna de proiectare cu Arduino si de familiarizare cu ceea ce presupune lucrul cu senzori, diferite module specializate, intreruperi hardware, protocoale de comunicatie specifice si cateva biblioteci.

Bibliografie/Resurse

- <https://ocw.cs.pub.ro/courses/pm/prj2010/amocanu/vocalcommand>
- <https://www.instructables.com/Arduino-Project-Baby-Noise-Detection/>
- <https://www.instructables.com/Music-Player-Using-Arduino/>
- <https://circuitdigest.com/microcontroller-projects/clap-switch-using-arduino>
- <https://learn.adafruit.com/adafruit-neopixel-uberguide/>
- <https://learn.adafruit.com/trinket-sound-reactive-led-color-organ>
- <https://learn.adafruit.com/adafruit-neopixel-uberguide/arduino-library-use>
- <https://ardushop.ro/ro/home/1314-modul-microfon-senzor-sunet.html>
- <https://electropeak.com/learn/interfacing-bt06-bluetooth-wireless-data-transmission-module-with-arduino/>
- https://www.youtube.com/watch?v=HQyqmpi-yLk&ab_channel=CrazyIndiana
- <https://forum.arduino.cc/t/sd-examples-and-chip-select-definition/126821/3>
- <https://circuitdigest.com/microcontroller-projects/arduino-audio-music-player>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/dene/jucariebebelusi>



Last update: **2023/05/30 11:04**