UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

AUTOMATIZAREA SISTEMULUI DE ILUMINARE ȘI SONORIZARE DIN LOCUINȚĂ

FACULTATEA: AUTOMATICĂ ȘI CALCULATOARE

SPECIALIZAREA: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

DISCIPLINA: MĂSURĂRI ELECTRONICE ȘI SENZORI

MORAR TIMOTEI CRISTIAN

Seria B, grupa 30225

CUPRINS

1.1 Domeniul de aplicabilitate

Dispozitivul proiectat prezintă mai multe funcționalități. Acesta a fost conceput la scară mică cu posibilități de extindere și îmbunătățire (a se vedea secțiunea 1.6 – Dezvoltare ulterioară). Acesta face controlul asupra becurilor, ledurilor și a mai multor dispozitive periferice mai ușor de gestionat, fiind posibil fie remotely prin simplul sunet produs de bătaia palmelor, fie dintr-un panou de comanda situat undeva la îndemâna utilizatorului.

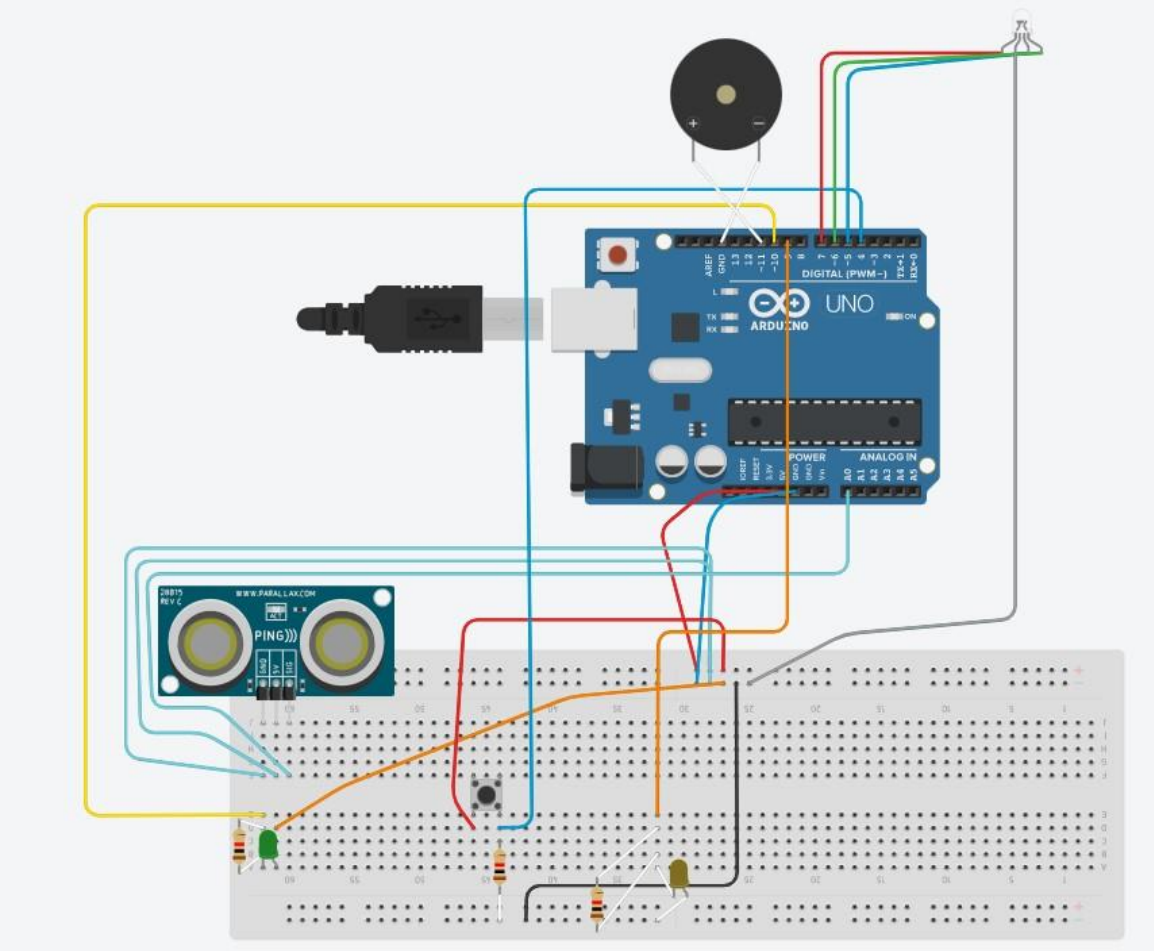
În viața cotidiană cred că cel mai mult aparatul este destinat copiilor încă speriați de frica de întuneric. Spre exemplu, atunci când eram mic și întrerupătorul de la scări era situat la baza scărilor, eu trebuia sa alerg repede pe scări de frica de a nu fi prins de cineva din întuneric. Mi-ar fi fost foarte de folos ca becurile să se stingă doar atunci când ajung sus pe scări și să bat de 2 ori din palme.

De asemenea, prin sistemul de sonorizare acesta se poate folosi spre exemplu pentru a acoperi toata casa într-o atmosferă de petrecere, de relaxare, etc. după bunul plac al utilizatorului, acesta fiind responsabil de muzica dorită.

1.2 Avantajul față de soluțiile cunoscute

Dispozitiv creat se bazează pe același principiu ca alte câteva produse aflate deja pe piață. Cu toate acestea consider că acesta conceput de mine prezintă un atu important în plus, și anume posibilitatea de a controla 2 sisteme de iluminare independente prin bătăi diferite, dar care să fie conectate la același microcontroler.

1.3 Schema de montaj și resursele utilizate



Tabel Pin-uri:

- *Modul Microfon*
  Analog Output : 2
  GND: GND
  VCC: 5V
  Digital Output: GND
- *Buzzer Piezo*
  +: 11
  -: GND
- *LED:*
  Verde: 10
  Galben: 9
  RGB: 7(roșu), 6(verde), 5(albastru)
- *Buton*
  +: VCC
  -: 4

Resurse:

1. Modul pentru detectarea sunetului de înaltă sensibilitate KY-037
2. LED-uri Verzi, Galbene și RGB
3. Buzzer Piezo
4. Arduino Uno
5. Breadboard
6. Fire
7. Buton de tip push
8. Rezistori

1.4 Principiu de funcționare și utilizare

Senzorul pentru detectarea sunetului KY-037 poate detecta energia mecanică a unei unde de sunet și o convertește într-un semnal electric. Aceste semnale pot fi amplificate, înregistrate, mixate, etc. Când unda de sunet lovește timpanul, se produce o vibrație în câmpul magnetic, creând astfel fluctuații în energia electrică.

Utilizarea dispozitivul prezintă următoarele funcționalități.

Dacă dorim aprinderea luminii naturale atunci tot ce avem de făcut este să batem de 2 ori consecvent din palme. Dacă observăm ca ledul nu s-a aprins, putem repeta din nou, iar dacă nici atunci nu se aprinde, este recomandat să se verifice microfonul modulului pentru ca nu cumva sa existe ceva care obstrucționează primirea de informație. În momentul în care se aprinde ledul un cântec sonor de cca. 2 secunde marchează pornirea ledului. Atâta timp cât nu se mai bate din palme ledul rămâne aprins. Daca se dorește stingerea becului trebuie doar sa batem din nou de 2 ori din palme. iar o melodie (cca. 2 sec) va marca stingerea ledului.

Daca dorim aprinderea luminii RGB, atunci batem o singură dată din palme, iar ledul se va aprinde, fără nicio notificare sonoră. Lumina care va apărea se va alege din întâmplare, datorită funcției random() din cod. Dacă vrem să schimbam culoarea batem o singură data din palme, iar aceasta își va schimba spectrul. Dacă dorim stingerea luminii RGB atunci batem de 2 ori din palme, ceea ce va porni lumina naturală, după 2 secunde batem din nou de 2 ori și lumina naturală se va stinge.

Pentru a porni sistemul de sonorizare tot ce avem de făcut este să apăsăm butonul de pe breadboard, iar un led se va porni care ne indică faptul ca muzica este redată. Din buzzer va ieși melodia de la Super Mario Game. Momentan aparatul nu are încorporate alte melodii. Dacă apăsăm din nou pe același buton muzica se va opri.

Dispozitivul nu necesită o locație precisă pentru a funcționa corespunzător. Are nevoie doar de o sursă de tensiune, deci fie conectat la o baterie de 9V fie conectat prin USB la un laptop/calculator.

Codul din Arduino aferent proiectului

```
const int Senzor = A0;            //pinul pentru senzorul care capteaza informatie
int clap2 = 0;                    //variabila care memoreaza daca am batut de 2 ori din
palme
long detection_range_start = 0;   //variabile pentru depistarea duratelor dintre batai
long detection_range = 0;
boolean status_lights = false;    //starea curenta a ledului
////////////////////////////////////////////////
const int buzzer = 11;            //pinul pentru buzzerul care difuzeaza sunetul
////////////////////////////////////////////////
const int red = 7;                //pinul RED din ledul RGB
const int green = 6;              //pinul GREEN din ledul RGB
const int blue = 5;               //pinul BLUE din ledul RGB
int clap1 = 0;                    //variabila care memoreaza daca am batut o data din
palme
long detection_range_start1 = 0;  //variabile pentru depistarea duratelor dintre batai
long detection_range1 = 0;
boolean status_lights1 = false;   //starea curenta a ledului RGB
////////////////////////////////////////////////
const int button=4;               //pinul pentru buton de PLAY/STOP
const int iesire=9;               //pinul pentru iesire si intrare in UNO
int buttonState=0;                //citirea ce vine din pinul pentru citire
int iesireState=LOW;              //starea curenta pentru pinul de intrare
int lastIesireState=LOW;          //starea precedenta pentru pinul de intrare


void setup() {
  pinMode(Senzor, INPUT);  //A0
  pinMode(10, OUTPUT);

  pinMode(red, OUTPUT); //7
  pinMode(green, OUTPUT); //6
  pinMode(blue, OUTPUT); //5

  pinMode(button,INPUT); //4
  pinMode(iesire,OUTPUT); //9
  digitalWrite(iesire,iesireState);
}

void loop() {
```

```
  aprindere_stingere_leduri();
  aprindereRGB();
  apasare_buton_play_stop();
}

void apasare_buton_play_stop(){
  unsigned long lastDebounceTime = 0; //variabila care ajuta la depistarea duratei de apasare a butonului
  unsigned long debounceDelay = 50;
  int reading = digitalRead(button); //reading ia valoarea HIGH daca butonul este apasat, altfel LOW

  if(reading!=lastIesireState) lastDebounceTime = millis();
  //daca am apasat butonul masuram timpul pana ridicam degetul de pe acesta, iar daca timpul depaseste 50 de ms(debounceDelay) este considerat VALID
  if ((millis() - lastDebounceTime) > debounceDelay) {
     if (reading != buttonState) {
      buttonState = reading; //transmitem mai departe informatia citita

      if (buttonState == HIGH){
        iesireState = !iesireState;
        super_mario();
      }
  }
  }
  digitalWrite(iesire,iesireState); //pe iesire afisam starea acesteia
  lastIesireState=reading; //si memoram ultima stare
}

void turning_on(){
  //subprogram care produce un anumit sunet atunci cand se bate de 2 ori pentru a porni ledul
  tone(buzzer, 783);
  delay(250);
  tone(buzzer, 987);
  delay(250);
  tone(buzzer, 1174);
  delay(250);
  tone(buzzer, 1566);
  delay(600);
  noTone(buzzer);
}

void turning_off(){
```

```
  //subprogram care produce un anumit sunet atunci cand se bate de 2 ori pentru a stinge ledul
  tone(buzzer, 1566);
  delay(250);
  tone(buzzer, 1174);
  delay(250);
  tone(buzzer, 987);
  delay(250);
  tone(buzzer, 783);
  delay(600);
  noTone(buzzer);
}

void aprindereRGB() {
  if (digitalRead(Senzor) == 0) {
    if (clap1 == 0) {
      detection_range_start1 = detection_range1 = millis(); //daca nu am patut deloc din palme incepem detectia bataii
      clap1++;
    } else if (clap1 > 0 && millis() - detection_range1 >= 50) {
      detection_range1 = millis();
      clap1++; //daca au trecut macar 50ms si am batut din palme atunci crestem numarul de batai
    }
  }
  if (millis() - detection_range_start1 >= 400) {
    if (clap1 == 1) { //in momentul cand avem o bataie nu ne mai intereseaza starea altor leduri, subprogramul alege random culorile si aprinde RGB-ul si stinge celalat led
        digitalWrite(10, LOW);
        analogWrite(red, random(0,255));
        analogWrite(green, random(0,255));
        analogWrite(blue, random(0,255));
        if(status_lights==true) turning_off();
        status_lights=false;
         }
    clap1 = 0;
  }
}

void aprindere_stingere_leduri() {
  if (digitalRead(Senzor) == 0) {
    if (clap2 == 0) {
      detection_range_start = detection_range = millis(); //daca nu am patut deloc din palme incepem detectia bataii
```

```
      clap2++;
    } else if (clap2 > 0 && millis() - detection_range >= 50) {
      detection_range = millis();
      clap2++; //daca au trecut macar 50ms si am batut din palme atunci crestem numarul
de batai
    }
  }
  if (millis() - detection_range_start >= 400) {
    if (clap2 == 2) { //in momentul in care am batut de 2 ori verificam daca ledul
verde(cel controleaza practic sistem) este stins/aprins
        digitalWrite(red, LOW);
        digitalWrite(green, LOW);
        digitalWrite(blue, LOW);
      if (!status_lights) {
        status_lights = true; //daca este stins il aprindem la urmatoare bataie de 2 ori
din palme si actualizam starea in 'APRINS'
        digitalWrite(10, HIGH);
        turning_on();
       } else if (status_lights) {
        status_lights = false; //daca este aprins il stingem la urmatoare bataie de 2
ori din palme si actualizam starea in 'STINS'
        digitalWrite(10, LOW);
        turning_off();
      }
    }
    clap2 = 0;
  }
}

void super_mario() {
  //Definim notele muzicale
  #define NOTE_E6 1319
  #define NOTE_G6 1568
  #define NOTE_A6 1760
  #define NOTE_AS6 1865
  #define NOTE_B6 1976
  #define NOTE_C7 2093
  #define NOTE_D7 2349
  #define NOTE_E7 2637
  #define NOTE_F7 2794
  #define NOTE_G7 3136
  #define NOTE_A7 3520

  //Linia melodica(partitura), unde este 0 inseamna ca este pauza
```

```
int melody[] = {
  NOTE_E7, NOTE_E7, 0, NOTE_E7,
  0, NOTE_C7, NOTE_E7, 0,
  NOTE_G7, 0, 0, 0,
  NOTE_G6, 0, 0, 0,

  NOTE_C7, 0, 0, NOTE_G6,
  0, 0, NOTE_E6, 0,
  0, NOTE_A6, 0, NOTE_B6,
  0, NOTE_AS6, NOTE_A6, 0,

  NOTE_G6, NOTE_E7, NOTE_G7,
  NOTE_A7, 0, NOTE_F7, NOTE_G7,
  0, NOTE_E7, 0, NOTE_C7,
  NOTE_D7, NOTE_B6, 0, 0,

  NOTE_C7, 0, 0, NOTE_G6,
  0, 0, NOTE_E6, 0,
  0, NOTE_A6, 0, NOTE_B6,
  0, NOTE_AS6, NOTE_A6, 0,

  NOTE_G6, NOTE_E7, NOTE_G7,
  NOTE_A7, 0, NOTE_F7, NOTE_G7,
  0, NOTE_E7, 0, NOTE_C7,
  NOTE_D7, NOTE_B6, 0, 0
};

//Duratele pentru fiecare dintre notele muzica de mai sus
int tempo[] = {
  12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,

  12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,

  9,9,9,12,12,12,12,12,12,12,12,12,12,12,12,

  12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,

  9,9,9,12,12,12,12,12,12,12,12,1212,12,12,
};

repetare1:
for (int thisNote = 0; thisNote < sizeof(melody) / sizeof(int); thisNote++) {
  if(iesireState==LOW) break; //conditia pentru oprire din cantat
  apasare_buton_play_stop();
```

```
    //Pentru a calcula durata unei note muzica, luam o secunda si o divizam la durata
notei
    //o patrime este 1000/4, o optime 1000/8, etc.

    int noteDuration = 1000 / tempo[thisNote];

    tone(buzzer, melody[thisNote], noteDuration);

    delay(noteDuration * 1.30);  //pauza intre note pentru a se putea diferentia acestea

    tone(buzzer, 0, noteDuration);
    if(thisNote==sizeof(melody)/sizeof(int)-1 && iesireState==HIGH) goto repetare1;
//daca nu s-a apasat pe buton ne intoarcem si repetam melodia inca o data:)
  }
}
```

1.5 Avantaje si dezavantaje

Consider că soluția aleasă este cea care se aproprie cel mai mult de realitate. Utilizarea aparatului este lizibilă, iar procesul de construcție și logica din spatele acestuia sunt ușor de înțeles, utilizând cunoștințe elementare din Circuite Analogice și Numerice si Măsurări Electronice și Senzori.

Aparatul este minimalist, compact și precis, săvârșind funcția desemnată. Principalul avantaj al acestuia este faptul că se pot controla 2 leduri conectate la același microcontroler prin gesturi diferite, fără a influența modul de funcționare unul față de altul.

Un dezavantaj îl reprezintă senzorul de sunet utilizat. Dacă cele 2 bătăi de palme nu se realizează separat una de cealaltă, senzorul nu va identifica bătăile ca 2 bătăi, ci ca o bătaie, ceea ce va aprinde ledul RGB, în locul ledului verde. Un alt dezavantaj îl reprezintă utilizarea buzzer-ului pe post de dispozitiv periferic. Acesta nu are cea mai bună funcționalitate ca difuzor.

1.6 Dezvoltare ulterioară

Proiectul prezentat este unul minimalist însă acesta se poate implementa în orice locuință. Spre exemplu, în locul ledului verde care semnalează aprinderea becurilor dintr-o încăpere putem conecta efectiv becurile la dispozitiv, însă avem nevoie de o sursă de tensiune mai mare, deoarece cea din plăcuța UNO nu este suficientă pentru a aprinde un bec.

De asemenea în locul utilizării ledului RGB, inițial dispozitivul a fost gândit pentru o banda LED care sa reacționeze la bătaia din palme.

Chiar mai mult putem adăuga mai multe melodii în memoria dispozitivul, și chiar și un buton prin care să alegem pe care dintre melodii dorim să le redam. Putem de asemenea să plasăm mai multe dispozitive periferice pentru un efect al sonorizării mai eficient.

Putem adăuga acțiuni diferite pentru diferite sunete produse.

1.7 Bibliografie

https://www.youtube.com/watch?v=Rf4MR27nNnw&t=307s&ab_channel=ElectronicClinic

https://www.princetronics.com/supermariothemesong/

https://www.youtube.com/@programmingelectronics