

HARP

Introduction

Brief presentation of your project:

- what does
- what is its purpose
- what was the idea you started from
- why you think it is useful for others and for you

General Description

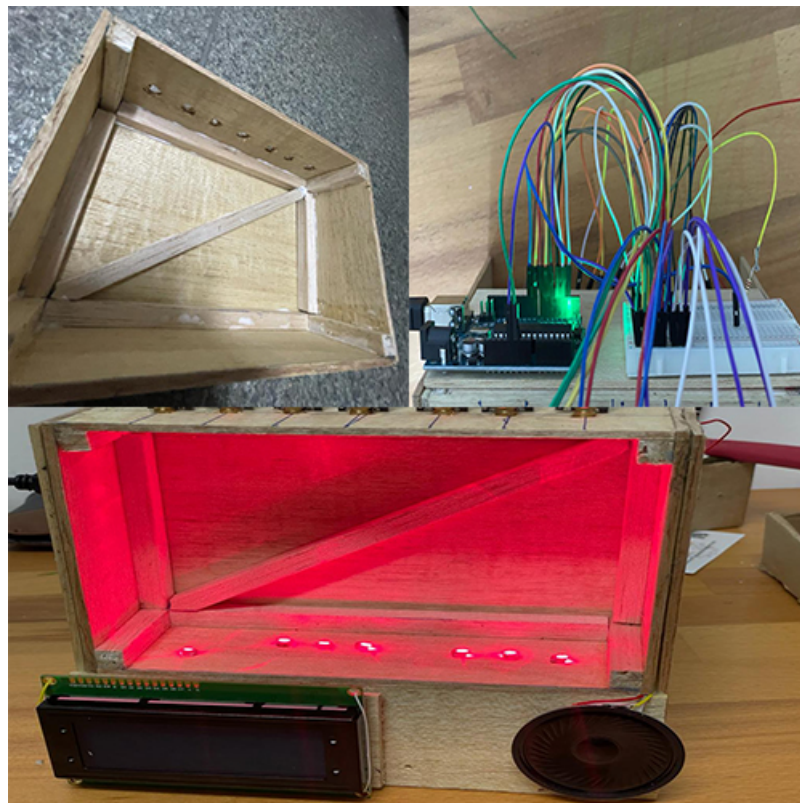
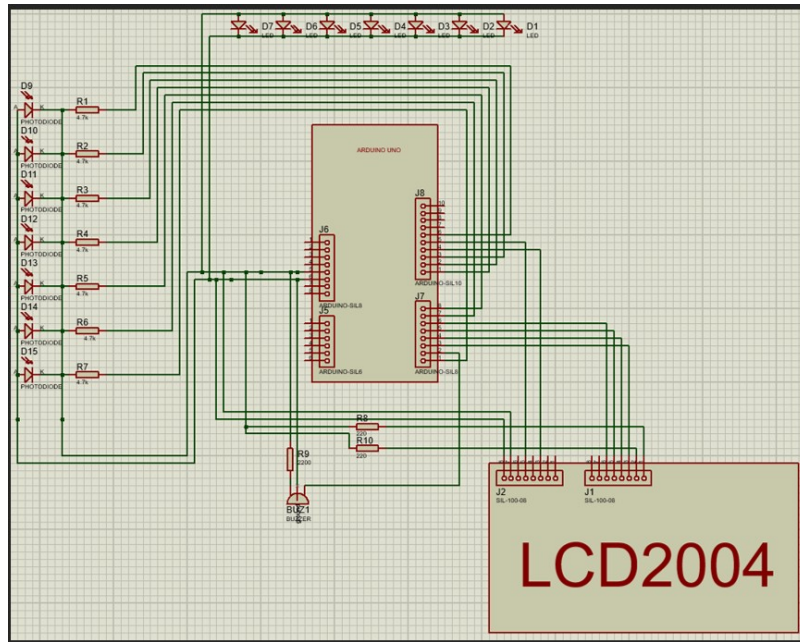
Introduction: The purpose of this project is to create an electrical harp composed of 7 lasers facing 7 detectors all connected to an Arduino with a speaker that will play a different song for each couple of lasers and detectors. We want to do this project because it's different than what we usually see as an Arduino project, and it can be interesting to enjoy building our own instrument with our own needs and desires. Description: The main principle is the following: when a detector detects a signal from the laser it's facing, nothing happens, but when it doesn't, that means someone put its finger on the « virtual laser chord » a song from an mp3 will be played with the right note. We are going to use a wood structure and we will try to make it have the shape of an actual harp. We are going to fix the lasers, detectors, the speaker, and the Arduino board on the wood to make one object like a real harp that you can take with you everywhere. We also want to implement an LCD screen that shows which note is being played when we turn on the harp. As a result, we are going to use everything we learned during lab 1, 3 and 4. Lab 3 because we are going to use a timer when we switch on the harp to trigger the note to be played, lab 4 because we are going to use an ADC with the detector and the laser.

Hardware Design

- 1 x Arduino Uno
- 9 x 50Ω photo-resistors
- 9 x 5mW red laser diodes

- 9 x 4.7K Ω resistor
- 1 x Breadboard
- LCD
- speaker
- cables

Schema Circuit



Software Design

```
#include <LiquidCrystal.h>

const int c = 262;
const int d = 294;
const int e = 330;
const int f = 349;
const int g = 392;
const int a = 440;
const int b = 494;

int kc = 0;
int kd = 0;
int ke = 0;
int kf = 0;
int kg = 0;
int ka = 0;
int kb = 0;

const int rs = 12;
const int en = 11;
const int d4 = 5;
const int d5 = 4;
const int d6 = 3;
const int d7 = 2;
const int contrastPin = A0;

const int lcdColumns = 20;
const int lcdRows = 4;

const int speakerPin = 9; // Broche de sortie audio

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

const int photodiodePin_c = 0;
const int photodiodePin_d = 1;
const int photodiodePin_e = 6;
const int photodiodePin_f = 7;
const int photodiodePin_g = 8;
const int photodiodePin_a = 10;
const int photodiodePin_b = A2;

void setup() {
  lcd.begin(lcdColumns, lcdRows);
```

```
        analogWrite(contrastPin, 75);
    lcd.print("Hello");
}

void loop() {

    bool laserDetected_c = digitalRead(photodiodePin_c);
    bool laserDetected_d = digitalRead(photodiodePin_d);
    bool laserDetected_e = digitalRead(photodiodePin_e);
    bool laserDetected_f = digitalRead(photodiodePin_f);
    bool laserDetected_g = digitalRead(photodiodePin_g);
    bool laserDetected_a = digitalRead(photodiodePin_a);
    bool laserDetected_b = digitalRead(photodiodePin_b);

    if (!laserDetected_c) {
        Serial.println("c detected");
        tone(speakerPin, c);
        kc = 1;
        delay(300);
        lcd.print("You are plaing a C !");
    }
    if (!laserDetected_d) {
        Serial.println("d detected");
        tone(speakerPin, d);
        kd = 1;
        delay(300);
        lcd.print("You are plaing a D !");
    }
    if (!laserDetected_e) {
        Serial.println("e detected");
        tone(speakerPin, e);
        ke = 1;
        delay(300);
        lcd.print("You are plaing a E !");
    }
    if (!laserDetected_f) {
        Serial.println("f detected");
        tone(speakerPin, f);
        kf = 1;
        delay(300);
        lcd.print("You are plaing a F !");
    }
    if (!laserDetected_g) {
        Serial.println("g detected");
        tone(speakerPin, g);
        kg = 1;
        delay(300);
        lcd.print("You are plaing a G !");
    }
    if (!laserDetected_a) {
```

```
    Serial.println("a detected");
    tone(speakerPin, a);
    ka = 1;
    delay(300);
    lcd.print("You are plaing an A !");
}
if (!laserDetected_b) {
    Serial.println("b detected");
    tone(speakerPin, b);
    kb = 1;
    delay(300);
    lcd.print("You are plaing a B !");
}

if (laserDetected_c && kc==1) {
    noTone(speakerPin);
    kc = 0;
    delay(300);
}
if (laserDetected_d && kd==1) {
    noTone(speakerPin);
    kd = 0;
    delay(300);
}
if (laserDetected_e && ke==1) {
    noTone(speakerPin);
    ke = 0;
    delay(300);
}
if (laserDetected_f && kf==1) {
    noTone(speakerPin);
    kf = 0;
    delay(300);
}
if (laserDetected_g && kg==1) {
    noTone(speakerPin);
    kg = 0;
    delay(300);
}
if (laserDetected_a && ka==1) {
    noTone(speakerPin);
    ka = 0;
    delay(300);
}
if (laserDetected_b && kb==1) {
    noTone(speakerPin);
    kb = 0;
    delay(300);
}
}
```


```
delay(100);  
  
}
```

Results Obtained

What were the results obtained after the realization of your project.

Conclusions

Download

An archive (or more if necessary) with the files obtained as a result of the project: sources, schemes, etc. A README file, a ChangeLog, a build script and automatic copy on uC always make a good impression .

The files are uploaded to the wiki using the **Add Images or other files** facility. The namespace in which the files are uploaded is of the type **:pm:prj20??:c?** or **:pm:prj20??:c?:student_name** (if applicable). **Example:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Journal

You can also have a journal section where the project assistant can track the progress of the project.

Bibliography/Resources

List of documents, datasheets, Internet resources used, possibly grouped by **Software Resources** and **Hardware Resources**.

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/apredescu/harp>



Last update: **2023/05/28 19:55**