

Piano a Laser

Victor Hugo Bezerra Tavares
FGA
Universidade de Brasília
Gama, Brasil
victorhugo.tavares@hotmail.com

Vanessa Oliveira Nóbrega
FGA
Universidade de Brasília
Gama, Brasil
vanessa.you.eng@gmail.com

Palavras-chave: *Raspberry, Hardware, Software.*

Desde que o bebê ainda está no ventre da mãe, a música já faz parte de sua vida. É através da tranquilidade que ela traz para a mãe que o pequeno é introduzido à sensibilização dos sons. O aprendizado de tocar um instrumento musical, traz uma série de benefícios importantes para o crescimento da criança, como o ganho de autoconfiança pois, a cada passo dado no aprendizado, a criança sente que pode fazer aquilo e passa a confiar mais em si mesma. O ganho de disciplina, pois para tocar um instrumento, é necessário bastante esforço e prática, o que necessita de muita disciplina. Assim, a criança passa a desenvolvê-la com mais frequência, como parte do seu dia a dia. O intuito do projeto seria então desenvolver um instrumento musical, no caso um piano, que de uma forma mais lúdica e atrativa auxiliaria a qualquer pessoa a desenvolver competências nesta área de atuação.

I. Desenvolvimento

A. Descrição de Hardware

Para a realização deste projeto foi utilizada a seguinte lista de materiais:

ITEM	QUANTIDADE
Raspberry Pi 3B	1
LDR (Light Dependent Resistor)	7
Capacitor Eletrolítico 1uF	7
LEDs(diodo emissor de luz)	42
Jumpers	-
Protoboard	1
Estrutura em MDF	-

O hardware consiste em uma matriz de led 7x6 e uma harpa a laser Fig. 1.

A matrix de led com cada uma das 7 colunas estará a frente de cada LDR, porque almejamos que fique claro para o utilizador qual feixe deva ser tocado. Devido a indisponibilidade de pinos GPIO suficientes da raspberry, para controle da matriz de led, foi utilizado um método chamado de charlieplexing que é uma técnica para a condução de um display multiplexado em que relativamente poucos dos pinos de I/O num microcontrolador são usados para conduzir uma matriz de LEDs, como pode ser visto na imagem 1 em anexo.

A Raspberry Pi possui somente saída e entrada digital, o que dificulta a leitura de sensores com dados de entrada analógicos como o LDR. Para fazer a leitura desse sensor, foi adotada a estratégia de construir um circuito RC com um capacitor em série com o sensor de luz, pois este coleta seus dados a partir da variação de sua resistência quando o mesmo é exposto a luz.

A estrutura foi feita em MDF (Medium-Density Fiberboard) e cortada com as medidas necessárias para comportar duas protoboards bem como os 7 lasers e os 7 LDRs.

B. Descrição de Software

Nesse ponto de entrega final de projeto fizemos algumas pequenas alterações no código dos LDRs, como adicionar os LDRs que faltavam para completar as 7 notas musicais e fizemos a otimização do código em python dos LEDs, e passamos ele para a linguagem C. Como pode ser visto no anexo 2 presente no final desse relatório.

No código (anexo 2) referente aos LDRs, é definido o pino GPIO como saída e como baixo, isso descarrega o capacitor. Em

seguida é definido o pino GPIO como entrada, aí a corrente flui através do resistor e do capacitor para o terra, a voltagem começa a subir em seguida. Esse tempo é proporcional a resistência do LDR. É monitorada o pino e lido seu valor é incrementado um contador. A tensão no capacitor aumenta o suficiente para ser considerada como HIGH pelo pino, por volta de 2v, o tempo gasto é proporcional a resistência do LDR e da quantidade de luz sobre ele. Em seguida é definido o pino como saída e repetido o processo.

Foi monitorado o tempo carga e descarga em cada LDR e feito uma condição para cada um.

Já na parte que estende os LEDs, foi feito uma condição para varrer os vetores referentes a matriz de LEDs, percorrendo assim uma ordem pré-estabelecida para reproduzir uma escala de luz que possibilita o usuário saber qual nota deveria ser tocada previamente para conseguir reproduzir uma musica.

II. Referências

- [1] Duarte, R. T. D. Harpa laser para controle de síntese sonora. Dissertação de mestrado. Faculdade de Engenharia da Universidade do Porto. 2011.
- [2] Freitas, D. G. G. Augustinho, J. G. Segecin, R. LaserHarpists: Harpa eletrônica com sons reproduzidos por dispositivo móvel. Relatório técnico. Universidade Tecnológica Federal do Paraná – UTFPR. 2015
- [3] Portal: ICExDuino. Disponível em: icexduino.blogspot.com.br/2011/09/projeto-harpa-laser.html. Acessado em: 02 de Abril de 2018.

[4] Portal: Instructables.
<http://www.instructables.com/id/Laser-Harp-with-Tah-and-Raspberry-Pi/>.
Acessado em: 01 de Abril de 2018.

Disponível em:

[5] Portal:

Anexo 1:

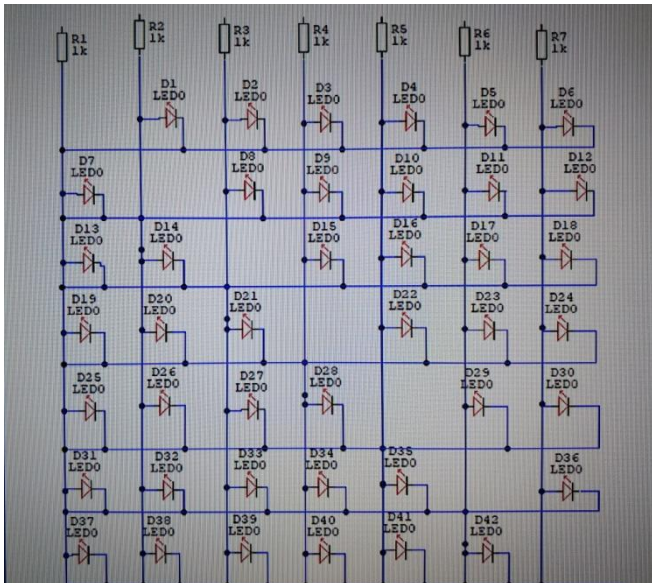


Imagem 1: Esquema de conexão charlieplexing da matriz 7x6

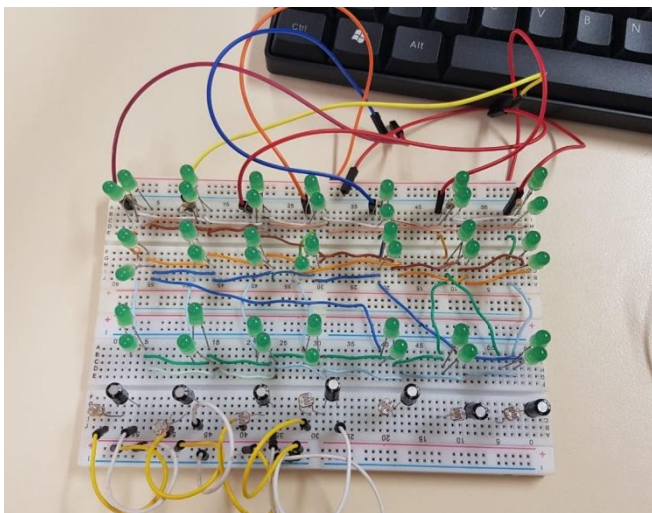


Imagem 2: Circuito apresentado no ponto de controle 4, contendo a matriz de LEDs sequencial e 4 LDRs que ao ser interrompida a emissão de luz sobre eles, o som de uma nota é emitida.

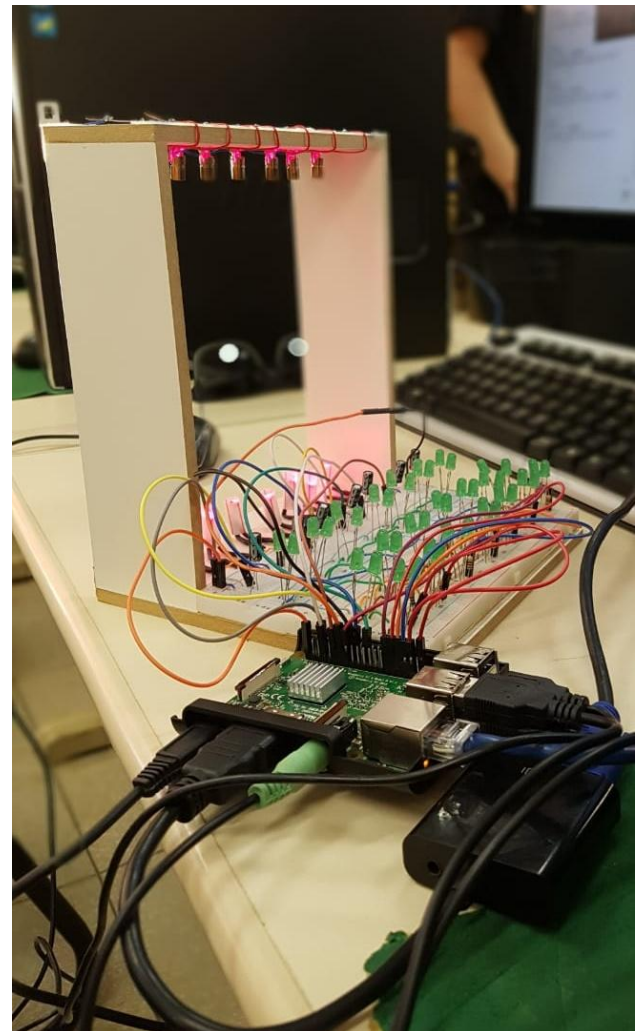


Imagem 3: Estrutura apresentada da Harpa Laser completa.

Anexo 2:

```

#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <alsa/asoundlib.h>
#include <alsa/pcm.h>

int main (void) {
    int pin,c1,c2,c3,c4,c5,c6,c7;
    char LDR;

    printf ("Raspberry Pi wiringPi Capacitor reading \n")
;

    if (wiringPiSetup () == -1)
        exit (1) ;

    for (pin = 0 ; pin < 8 ; ++pin) {
        pinMode (pin, OUTPUT) ;
        digitalWrite (pin, LOW) ;
    }

    for (;;) {
        pinMode (1, OUTPUT);
        digitalWrite (1, LOW);
        delay(50);
        c1=0;
        pinMode (1, INPUT);
        while (digitalRead(1)==LOW)
            c1++;

        pinMode (4, OUTPUT);
        digitalWrite (4, LOW);
        delay(50);
        c2=0;
        pinMode (4, INPUT);
        while (digitalRead(4)==LOW)
            c2++;

        pinMode (5, OUTPUT);
        digitalWrite (5, LOW);
        delay(50);
        c3=0;
        pinMode (5, INPUT);
        while (digitalRead(5)==LOW)
            c3++;

        pinMode (6, OUTPUT);

```

```

        digitalWrite (6, LOW);
        delay(50);
        c4=0;
        pinMode (6, INPUT);
        while (digitalRead(6)==LOW)
            c4++;

        if (c1 > 12000){
            system("aplay piano-a.wav");
        }

        else
            if (c2 > 26000){
                system("aplay piano-b.wav");
            }

        else
            if (c3 > 30000){
                system("aplay piano-c.wav");
            }

        else
            if (c4 > 70000){
                system("aplay piano-d.wav");
            } } }

#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>

```

```
#include <time.h>
```

```
#define pin1 4
```

```
#define pin2 17
```

```
#define pin3 27
```

```
#define pin4 22
```

```
#define pin5 5
```

```
#define pin6 6
```

```
#define pin7 13
```

```
int H = 1
```

```
int L = 0
```

```
int O = -1
```

```
int LEDS [256][7];
```

```
signed int LEDS={ {H, L, O, O, O, O, O},
```

```
    {H, O, L, O, O, O, O},
```

```
    {H, O, O, L, O, O, O},
```

```
    {H, O, O, O, L, O, O},
```

```
    {H, O, O, O, O, L, O},
```

```
    {H, O, O, O, O, O, L},
```

```
    {L, H, O, O, O, O, O},
```

```
    {O, H, L, O, O, O, O},
```

```
    {O, H, O, L, O, O, O},
```

```
    {O, H, O, O, L, O, O},
```

```
    {O, H, O, O, O, L, O},
```

```
    {O, H, O, O, O, O, L},
```

```
    {L, O, H, O, O, O, O},
```

```
    {O, L, H, O, O, O, O},
```

```
    {O, O, H, L, O, O, O},
```

```
    {O, O, H, O, L, O, O},
```

```
    {O, O, H, O, O, L, O},
```

```
    {O, O, H, O, O, O, L},
```

```
    {L, O, O, H, O, O, O},
```

```
    {O, L, O, H, O, O, O},
```

```
    {O, O, L, H, O, O, O},
```

```
    {O, O, O, H, L, O, O},
```

```
    {O, O, O, H, O, L, O},
```

```
    {O, O, O, H, O, O, L},
```

```
    {L, O, O, O, H, O, O},
```

```
    {O, L, O, O, H, O, O},
```

```
    {O, O, L, O, H, O, O},
```

```
    {O, O, O, L, H, O, O},
```

```
    {O, O, O, O, H, L, O},
```

```
    {O, O, O, O, H, O, L},
```

```
    {L, O, O, O, O, H, O},
```

```
    {O, L, O, O, O, H, O},
```

```
    {O, O, L, O, O, H, O},
```

```
    {O, O, O, L, O, H, O},
```

```
    {O, O, O, O, L, H, O},
```

```
    {O, O, O, O, O, H, L},
```

```
    {L, O, O, O, O, O, H},
```

```
    {O, L, O, O, O, O, H},
```

```
    {O, O, L, O, O, O, H},
```

```
    {O, O, O, L, O, O, H},
```

```
    {O, O, O, O, L, O, H},
```

```
    {O, O, O, O, O, L, H}}
```

```
int main(void){
```

```
while(1){
```

```
    for( i = 0 ; i <= 255; i++){
```

```
        for( j = 0 ; j <= 6; j++){
```

```
            if{
```

```
                LEDS[i][j] == O;
```

```
                wiringPiSetup();
```

```
                pinMode(LEDS[i][j], INPUT);
```

```
            }else{ // se não
```

```
                if LEDS[i][j] == H{
```

```
                    wiringPiSetup();
```

```
                    pinMode(LEDS[i][j], OUTPUT);
```

```
                    digitalWrite (pin, LOW) ;
```

```
                }else{
```

```
                    if LEDS[i][j] == L{
```

```
                        wiringPiSetup();
```

```
                        pinMode(LEDS[i][j], OUTPUT);
```

```
                        digitalWrite (pin, LOW) ;
```

```
                    } } } }
```

```
                delay(90);
```

```
            } } }
```

```
        GPIO.cleanup()
```

```
    return(0);
```

```
}
```

