

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
ENGENHARIA DE COMPUTAÇÃO**

PROJETO SCATMAN'S DIMMER

**CURITIBA
2010**

**GILBERTO YOSHIKI YAMANOUCHI
RAFAEL HENRIQUE AZANHA DE ORNELAS
JORGE HENRIQUE WERNECK GOMES
THIAGO GONÇALVES CLASSEN**

PROJETO SCATMAN'S DIMMER

Projeto apresentado ao curso de Engenharia de Computação do Centro de Ciências Exatas e Tecnológicas da Pontifícia Universidade Católica do Paraná como critério de avaliação de Microprocessadores I, sob a orientação do Prof. Afonso Ferreira Miguel.

**CURITIBA
2010**

SUMÁRIO

INTRODUÇÃO	3
JUSTIFICATIVAS	4
OBJETIVO	5
METODOLOGIA	6
DESENVOLVIMENTO	7
CÓDIGO DO PROGRAMA	8
MATERIAIS UTILIZADOS	13
CONCLUSÃO	14
GLOSSÁRIO	15
REFERÊNCIAS	17
ANEXOS – FOTOS	18

INTRODUÇÃO

O projeto desenvolvido durante o quinto período do Curso de Engenharia de Computação consiste em um sistema de automação de iluminação para shows. O protótipo do sistema é representado por uma maquete de um palco iluminado por leds que acendem automaticamente de acordo com sinais MIDI recebidos de qualquer programa de edição de música com saída serial.

Estes sinais são recebidos pelo circuito e chegam ao programa. Para cada nível de sinal – nota MIDI – existe um padrão pré-definido no programa indicando a ordem em que os leds devem acender.

JUSTIFICATIVAS

Existem vários tipos de controle de iluminação disponíveis. A maioria utiliza controladores DMX e computadores para os canais de entrada. Porém, o preço deste tipo de equipamento é elevado e demanda operação manual.

Este projeto tem como diferencial a automação, porém com padrões programados previamente, tendo um controle preciso por tempo de execução da música.

OBJETIVO

Este projeto tem como objetivo didático a integração e ampliação dos conhecimentos adquiridos durante o curso, enfatizando as matérias de eletrônica e microprocessadores.

Além da parte técnica que o grupo deve realizar em software e hardware, implementando conhecimentos práticos e teóricos, outros elementos como planejamento, organização, trabalho em equipe e divisão de tarefas dentro de cronogramas são realizados durante o projeto.

METODOLOGIA

Para agilizar a construção do projeto foi adquirido Arduino, uma plataforma eletrônica open-source ideal para receber e enviar dados e atuar através motores e leds, criando facilmente objetos interativos interessantes.

Enquanto a maquete estava sendo feita se iniciaram pesquisas em turno do Arduino para saber o que existia de bibliotecas prontas e o que já havia sido feito de parecido com nosso projeto.

Após a aquisição do cabo USB/MIDI e estudos sobre como a comunicação com sinais MIDI funciona se iniciaram os testes em protoboard utilizando o programa Sonar para rodar a música.

Com os padrões definidos no programa e a maquete do palco pronta, foram feitos os circuitos com leds. Por fim, a união de todos os elementos do projeto, testes e ajustes finais.

DESENVOLVIMENTO

Com a aquisição do Arduino foi iniciada sua programação. Adicionando-se uma biblioteca MIDI para a recepção de dados do computador fez-se possível programar os padrões com os quais os leds são acesos.

A partir do programa para edição de áudio Sonar (Cakewalk), um sinal MIDI é enviado através do cabo USB-MIDI para o Arduino, o qual traduz os sinais recebidos através dos padrões pré-programados, deste modo faz-se necessário uma configuração MIDI única para cada música desejada.

Nas saídas do Arduino estão conectados os circuitos responsáveis pela alimentação dos leds tanto da parte superior, quanto da parte inferior do palco, utilizando um total de 9 saídas para realizar a iluminação.

No espaço interno do palco, está presente um sensor de luz (LDR), juntamente com um led para a referência do mesmo. Este sensor é responsável pela ativação das luzes de ambiente (LED Branco), fixado na lateral esquerda do palco e deve acender quando o led de referência for apagado, ou seja, ao término da música.

CÓDIGO DO PROGRAMA

```
#include <MIDI.h>

//variaveis LDR
int valor = 0; // Variavel onde e guardado o valor lido do LDR
int led_pin = 13; // Pino onde vamos ligar o LED
int analogin = 0; // Pino onde vamos ligar a LDR
//fim variaveis LDR
//variaveis LED MIDI
int pin2 = 2;
int pin3 = 3;
int pin4 = 4;
int pin5 = 5;
int pin6 = 6;
int pin7 = 7;
int pin8 = 8;
int pin9 = 9;
int pin10 = 10;
//fim variaveis LED MIDI
void HighLed(byte num)
{
  if(num>=0 &&num <8)
  {
    digitalWrite(pin10, HIGH); //LED de base para o LDR,
    digitalWrite(pin2, HIGH);
    digitalWrite(pin3, LOW);
    digitalWrite(pin4, LOW);
    digitalWrite(pin5, LOW);
    digitalWrite(pin6, LOW);
    digitalWrite(pin7, LOW);
    digitalWrite(pin8, LOW);
    digitalWrite(pin9, LOW);
    delayMicroseconds(50);
  }
  else if(num>=8 && num <16)
  {
    digitalWrite(pin10, HIGH); //LED de base para o LDR,
    digitalWrite(pin2, LOW);
    digitalWrite(pin3, HIGH);
    digitalWrite(pin4, LOW);
    digitalWrite(pin5, LOW);
    digitalWrite(pin6, LOW);
    digitalWrite(pin7, LOW);
    digitalWrite(pin8, LOW);
    digitalWrite(pin9, LOW);
    delayMicroseconds(50);
  }
  else if(num>=16 && num <24)
  {
    digitalWrite(pin10, HIGH); //LED de base para o LDR,
    digitalWrite(pin2, LOW);
    digitalWrite(pin3, LOW);
    digitalWrite(pin4, HIGH);
    digitalWrite(pin5, LOW);
    digitalWrite(pin6, LOW);
    digitalWrite(pin7, LOW);
    digitalWrite(pin8, LOW);
    digitalWrite(pin9, LOW);
    delayMicroseconds(50);
  }
  else if(num>=24 && num <32)
```

```

{
  digitalWrite(pin10, HIGH); //LED de base para o LDR,
  digitalWrite(pin2, HIGH);
  digitalWrite(pin3, HIGH);
  digitalWrite(pin4, LOW);
  digitalWrite(pin5, LOW);
  digitalWrite(pin6, LOW);
  digitalWrite(pin7, LOW);
  digitalWrite(pin8, LOW);
  digitalWrite(pin9, LOW);
  delayMicroseconds(50);
}
else if(num>=32 && num <40)
{
  digitalWrite(pin10, HIGH); //LED de base para o LDR,
  digitalWrite(pin2, HIGH);
  digitalWrite(pin3, HIGH);
  digitalWrite(pin4, HIGH);
  digitalWrite(pin5, LOW);
  digitalWrite(pin6, LOW);
  digitalWrite(pin7, LOW);
  digitalWrite(pin8, LOW);
  digitalWrite(pin9, LOW);
  delayMicroseconds(50);
}
else if(num>=40 && num <48)
{
  digitalWrite(pin10, HIGH); //LED de base para o LDR,
  digitalWrite(pin2, LOW);
  digitalWrite(pin3, HIGH);
  digitalWrite(pin4, HIGH);
  digitalWrite(pin5, LOW);
  digitalWrite(pin6, LOW);
  digitalWrite(pin7, LOW);
  digitalWrite(pin8, LOW);
  digitalWrite(pin9, LOW);
  delayMicroseconds(50);
}
else if(num>=48 && num <56)
{
  digitalWrite(pin10, HIGH); //LED de base para o LDR,
  digitalWrite(pin2, HIGH);
  digitalWrite(pin3, LOW);
  digitalWrite(pin4, LOW);
  digitalWrite(pin5, HIGH);
  digitalWrite(pin6, LOW);
  digitalWrite(pin7, LOW);
  digitalWrite(pin8, LOW);
  digitalWrite(pin9, HIGH);
  delayMicroseconds(50);
}
else if(num>=56 && num <64)
{
  digitalWrite(pin10, HIGH); //LED de base para o LDR,
  digitalWrite(pin2, HIGH);
  digitalWrite(pin3, LOW);
  digitalWrite(pin4, LOW);
  digitalWrite(pin5, HIGH);
  digitalWrite(pin6, HIGH);
  digitalWrite(pin7, LOW);
  digitalWrite(pin8, HIGH);
  digitalWrite(pin9, HIGH);
}

```

```

    delayMicroseconds(50);
}
else if(num>=64 && num <72)
{
    digitalWrite(pin10, HIGH); //LED de base para o LDR,
    digitalWrite(pin2, HIGH);
    digitalWrite(pin3, LOW);
    digitalWrite(pin4, LOW);
    digitalWrite(pin5, HIGH);
    digitalWrite(pin6, HIGH);
    digitalWrite(pin7, HIGH);
    digitalWrite(pin8, HIGH);
    digitalWrite(pin9, HIGH);
    delayMicroseconds(50);
}
else if(num>=72 && num <80)
{
    digitalWrite(pin10, HIGH); //LED de base para o LDR,
    digitalWrite(pin2, LOW);
    digitalWrite(pin3, HIGH);
    digitalWrite(pin4, LOW);
    digitalWrite(pin5, LOW);
    digitalWrite(pin6, HIGH);
    digitalWrite(pin7, HIGH);
    digitalWrite(pin8, HIGH);
    digitalWrite(pin9, LOW);
    delayMicroseconds(50);
}
else if(num>=80 && num <88)
{
    digitalWrite(pin10, HIGH); //LED de base para o LDR,
    digitalWrite(pin2, LOW);
    digitalWrite(pin3, HIGH);
    digitalWrite(pin4, LOW);
    digitalWrite(pin5, LOW);
    digitalWrite(pin6, LOW);
    digitalWrite(pin7, HIGH);
    digitalWrite(pin8, LOW);
    digitalWrite(pin9, LOW);
    delayMicroseconds(50);
}
else if(num>=88 && num <96)
{
    digitalWrite(pin10, HIGH); //LED de base para o LDR,
    digitalWrite(pin2, LOW);
    digitalWrite(pin3, LOW);
    digitalWrite(pin4, HIGH);
    digitalWrite(pin5, LOW);
    digitalWrite(pin6, LOW);
    digitalWrite(pin7, LOW);
    digitalWrite(pin8, LOW);
    digitalWrite(pin9, LOW);
    delayMicroseconds(50);
}
else if(num>=96 && num <104)
{
    digitalWrite(pin10, HIGH); //LED de base para o LDR,
    digitalWrite(pin2, HIGH);
    digitalWrite(pin3, LOW);
    digitalWrite(pin4, HIGH);
    digitalWrite(pin5, LOW);
    digitalWrite(pin6, HIGH);

```

```

    digitalWrite(pin7, LOW);
    digitalWrite(pin8, HIGH);
    digitalWrite(pin9, LOW);
    delayMicroseconds(50);
}
else if(num>=104 && num <112)
{
    digitalWrite(pin10, HIGH); //LED de base para o LDR,
    digitalWrite(pin2, HIGH);
    digitalWrite(pin3, LOW);
    digitalWrite(pin4, HIGH);
    digitalWrite(pin5, HIGH);
    digitalWrite(pin6, LOW);
    digitalWrite(pin7, HIGH);
    digitalWrite(pin8, LOW);
    digitalWrite(pin9, HIGH);
    delayMicroseconds(50);
}
else if(num>=112 && num <120)
{
    digitalWrite(pin10, HIGH); //LED de base para o LDR,
    digitalWrite(pin2, LOW);
    digitalWrite(pin3, LOW);
    digitalWrite(pin4, LOW);
    digitalWrite(pin5, LOW);
    digitalWrite(pin6, LOW);
    digitalWrite(pin7, LOW);
    digitalWrite(pin8, LOW);
    digitalWrite(pin9, LOW);
    delayMicroseconds(50);
}
else if(num >= 120)
{
    digitalWrite(pin10, LOW); //LED de base para o LDR,
    digitalWrite(pin2, LOW);
    digitalWrite(pin3, LOW);
    digitalWrite(pin4, LOW);
    digitalWrite(pin5, LOW);
    digitalWrite(pin6, LOW);
    digitalWrite(pin7, LOW);
    digitalWrite(pin8, LOW);
    digitalWrite(pin9, LOW);
    delayMicroseconds(50);
}
}
void setup()
{
    Serial.begin(9600);
    //setup LDR
    pinMode(led_pin, OUTPUT);
    //fim setup LDR

    //setup LED MIDI
    pinMode(pin2,OUTPUT);
    pinMode(pin3,OUTPUT);
    pinMode(pin4,OUTPUT);
    pinMode(pin5,OUTPUT);
    pinMode(pin6,OUTPUT);
    pinMode(pin7,OUTPUT);
    pinMode(pin8,OUTPUT);
    pinMode(pin9,OUTPUT);
    pinMode(pin10,OUTPUT);
}

```

```

MIDI.begin();
//fim setup LED MIDI
}

void loop()
{
  //loop LDR
  valor = analogRead(analogin); // O valor que irá ser lido na porta analogica numero 6 irá ser guardado na
  variavel "valor"
  if (valor < 360)
  { // Se a variavel valor for menor que 300
    digitalWrite(led_pin, HIGH); // acende o led da porta 8
  }
  else if(valor>450)
  {
    digitalWrite(led_pin, LOW);
  }
  //fim loop LDR

  //loop LED MIDI
  if(MIDI.read())
  {
    switch(MIDI.getType())
    {
      case NoteOn:
        Serial.print(MIDI.getData1(),BYTE);
        Serial.println("");
        HighLed(MIDI.getData1());
        break;
      default:
        break;
    }
  }
  //fim loop LED MIDI
}

```

MATERIAIS UTILIZADOS

- 1 Cabo MIDI/USB;
- 5 Resistores de 10 ohms;
- 5 Resistores de 100 ohms;
- 1 Resistor de 220 ohms;
- 1 Resistor de 3.3k ohms;
- 1 Resistor de 100k ohms;
- 1 Led Rosa;
- 2 Leds Amarelos;
- 2 Leds Violetas;
- 2 Leds Azuis;
- 2 Leds Verdes;
- 2 Leds Vermelhos;
- 2 Leds Brancos;
- 1 LDR;
- 1 4N35;
- 1 ULN2803
- Arduino;
- MDF;
- Notebook com o programa Sonar ou equivalente instalado. Necessário para rodar o arquivo MIDI;

CONCLUSÃO

Com o fim deste projeto concluiu-se que controlar um jogo de luzes a partir de arquivos MIDI editados previamente não é a melhor opção, tanto pelo trabalho de edição levando em conta alterações de luzes em um determinado espaço de tempo muito curto quanto por problemas que o programa que roda o arquivo MIDI pode causar com atrasos e até mesmo travar o funcionamento do sistema. Também que deveria ser feito um controle elétrico melhor para os leds a fim de usar mais destes com a corrente elétrica disponível do circuito.

Para o grupo, fica a experiência de aplicar na prática os conhecimentos adquiridos com os professores, o tratamento de sinais MIDI e a programação de microprocessadores, mesmo que através do Arduino.

GLOSSÁRIO

MIDI

(Musical Instrument Digital Interface), ou Interface Digital para Instrumentos Musicais, é uma tecnologia padronizada de comunicação entre instrumentos musicais e equipamentos eletrônicos possibilitando que uma composição musical seja executada, transmitida ou manipulada por qualquer dispositivo que reconheça esse padrão. Tecnicamente, MIDI é um protocolo; entretanto, o termo geralmente é utilizado também para se referir aos diversos componentes do sistema, como adaptadores, conectores, arquivos, cabos etc.

USB

Universal Serial Bus (USB) é um tipo de conexão "*ligar e usar*" que permite a conexão de periféricos sem a necessidade de desligar o computador.

LED

O LED é um diodo semicondutor (junção P-N) que quando energizado emite luz visível por isso LED (Diodo Emissor de Luz).

Arduino

Computador físico baseado numa simples plataforma de hardware livre, projetada com um microcontrolador de placa única, com suporte de entrada/saída embutido e uma linguagem de programação padrão, na qual tem origem em *Wiring*, e é essencialmente C/C++.

Sonar

Programa de alto nível para edição profissional de música com entradas e saídas para porta serial.

Resistor

Resistor é um dispositivo elétrico muito utilizado em eletrônica, com a finalidade de transformar energia elétrica em energia térmica (efeito joule), a partir do material empregado, que pode ser por exemplo carbono ou silício.

LDR

Light Dependent Resistor ou em português *Resistor Dependente de Luz* é um tipo de resistor cuja resistência varia conforme a intensidade de radiação eletromagnética do espectro visível que incide sobre ele.

REFERÊNCIAS

AllDatasheet – Datasheet Search Site

<http://www.alldatasheet.com>

Arduino

<http://www.arduino.cc>

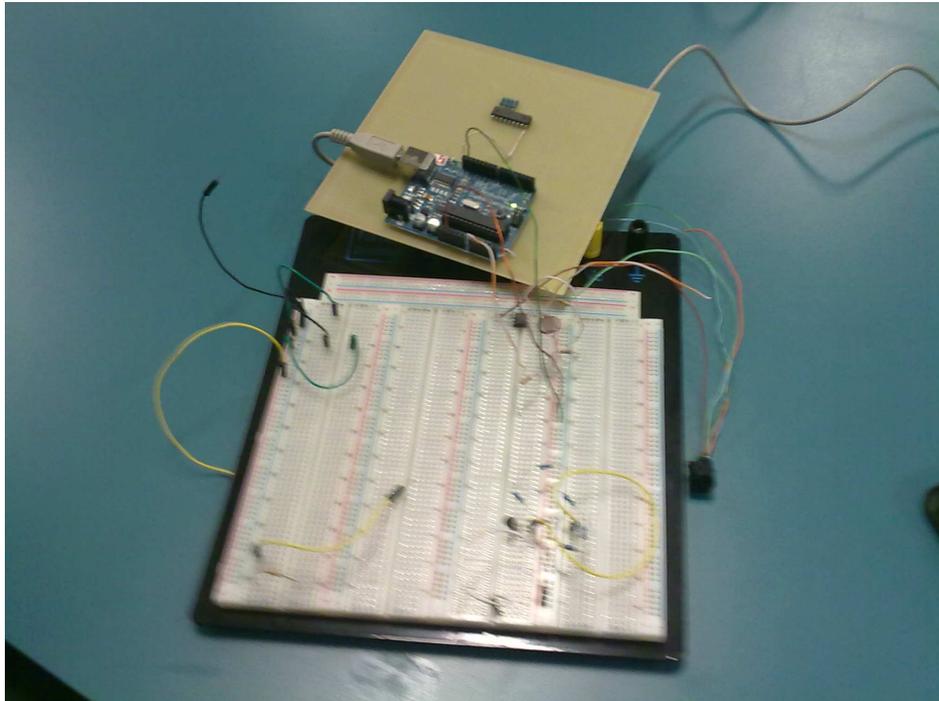
Arduino Brasil

<http://www.arduino.com.br>

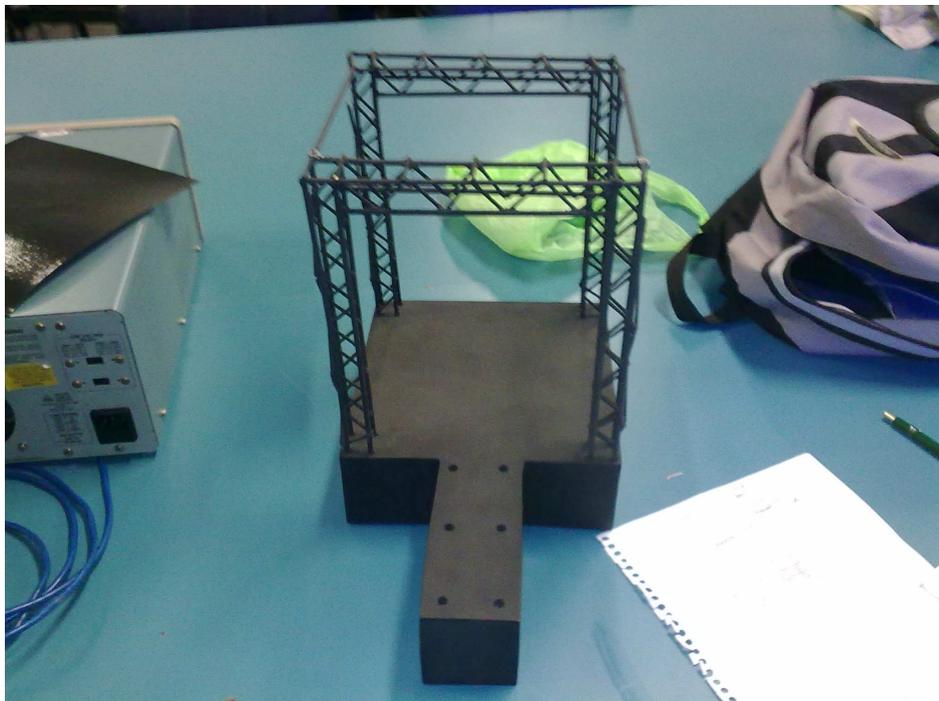
Wikipedia

<http://pt.wikipedia.org>

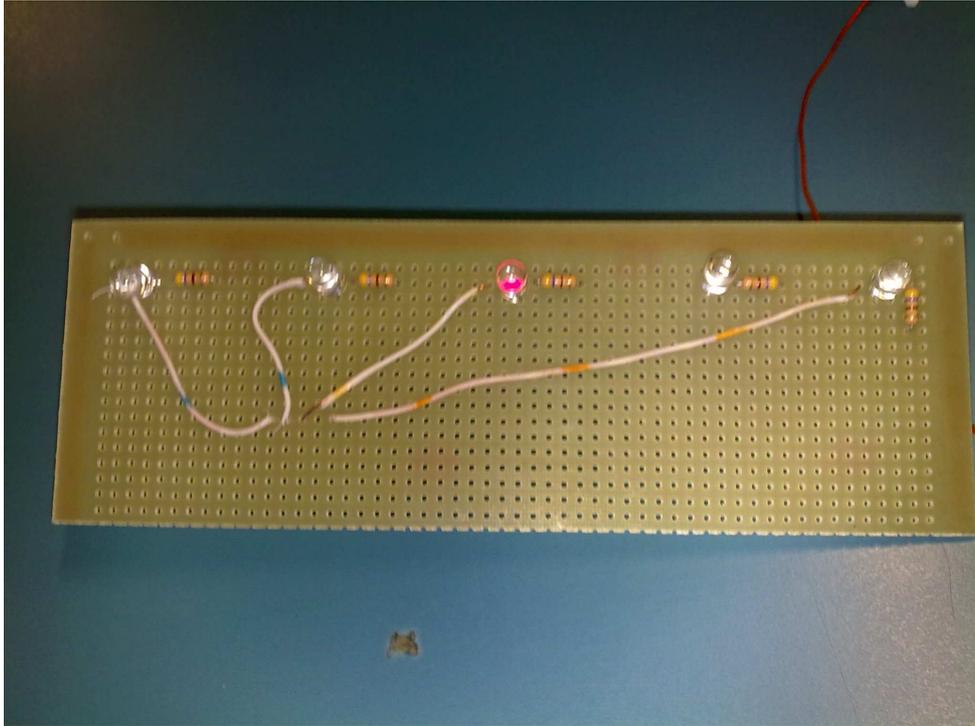
ANEXOS – FOTOS



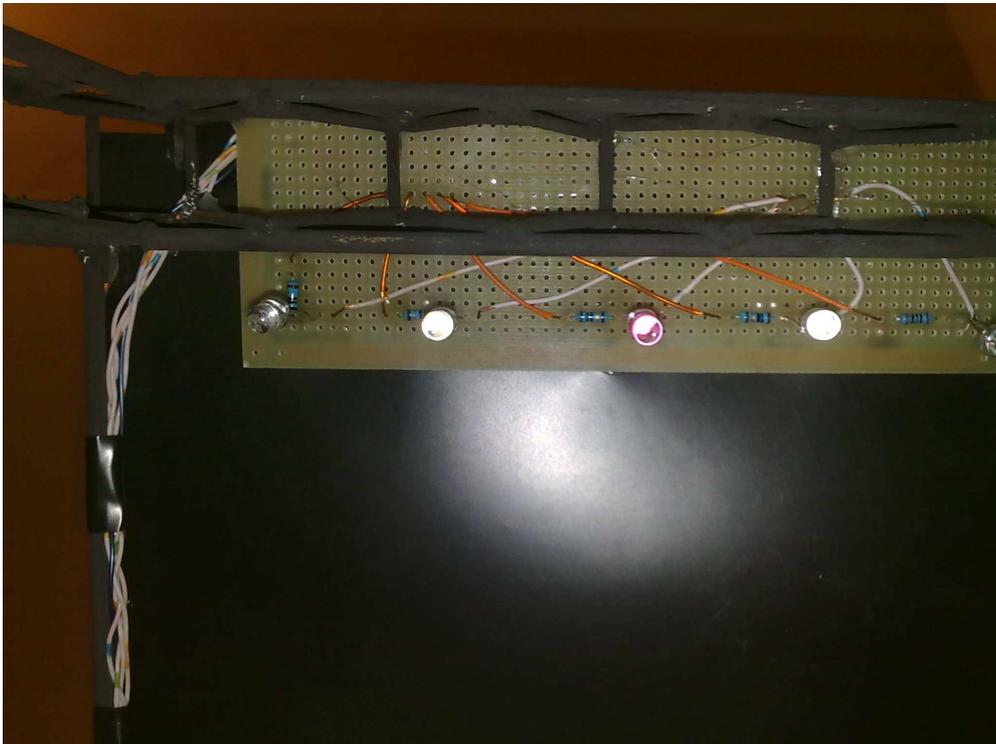
Testes com Arduino no protoboard



Estrutura do palco



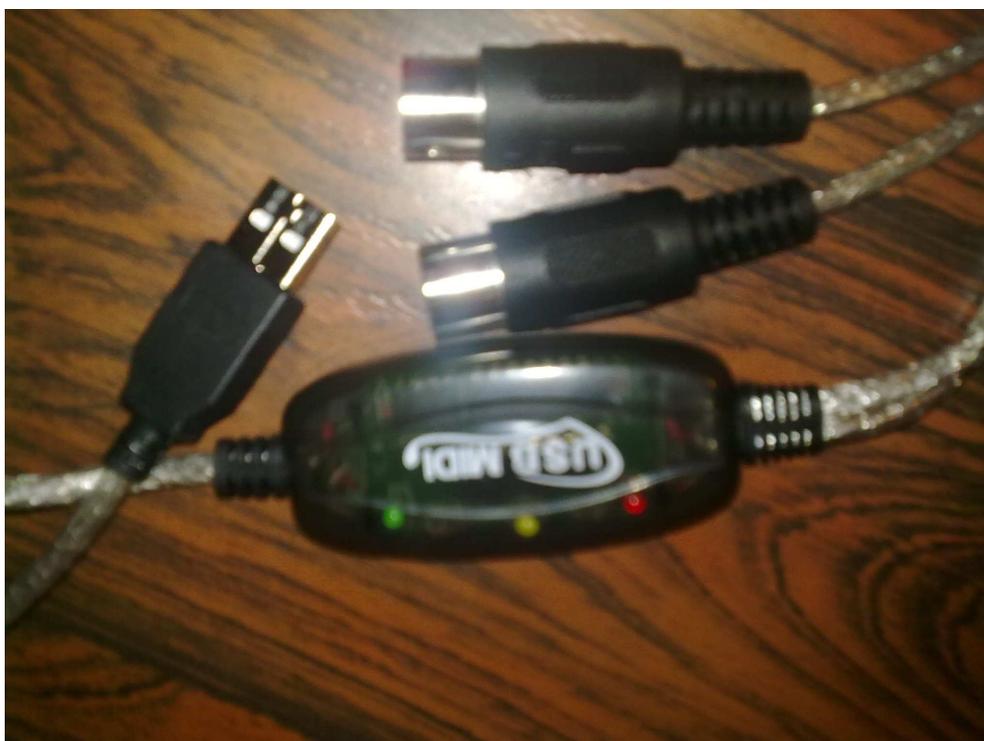
Placa com montagem dos leds



Placa com leds na estrutura



Saída do LDR para o led na lateral



Cabo USB/MIDI