

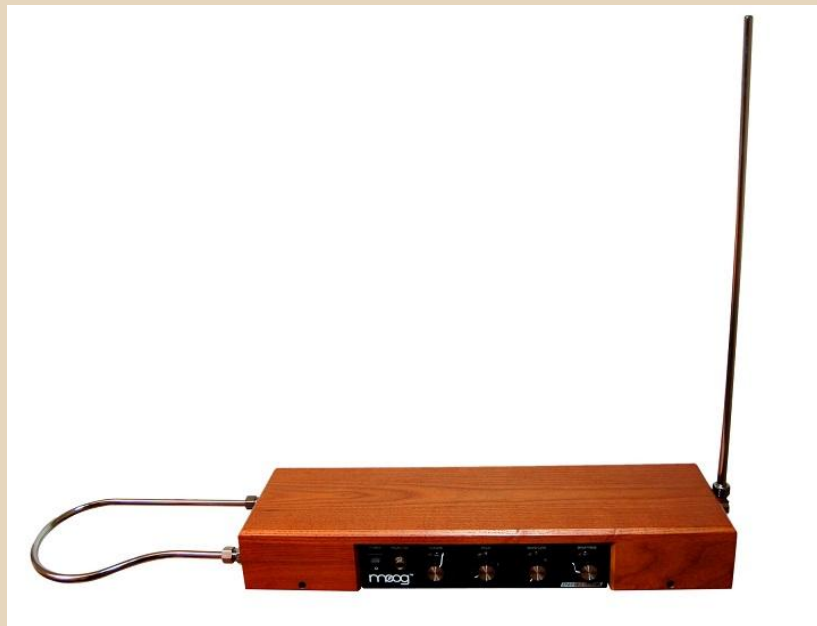
Construção de Theremin com sensores fotoelétricos

João Luiz Grave Gross

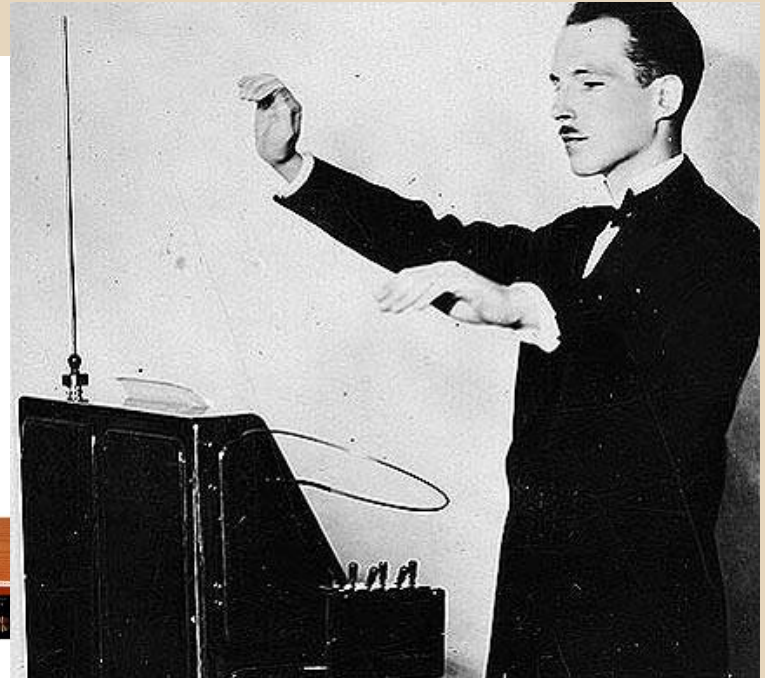
O Theremin - Histórico

- Instrumento musical eletrônico criado pelo russo Lev Sergeivitch Termen
 - Físico e ex-agente da KGB
 - Desenvolvimento dentro de um programa de pesquisa de sensores de proximidade financiado pelo governo russo até 1920
 - Patente realizada apenas em 1928
- A ideia: percepção da interferência do som em um aparelho de rádio ao aproximar a mão da antena.
 - Caixa de válvulas + duas antenas, uma horizontal e uma vertical
 - Ao aproximar as mãos altera-se o campo eletromagnético das antenas

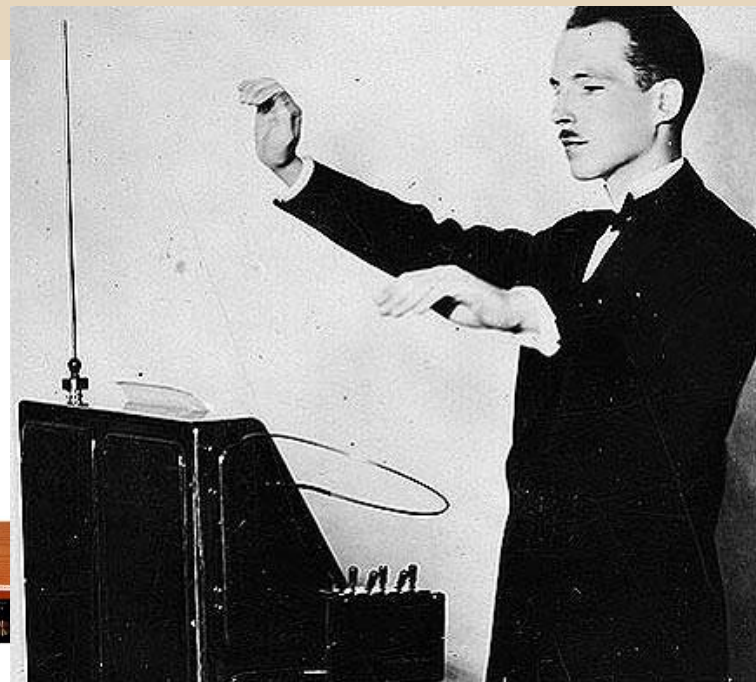
O Theremin - Aparência



O Theremin - Aparência



O Theremin - Aparência



O Theremin - Como tocar?



Proposta do Trabalho

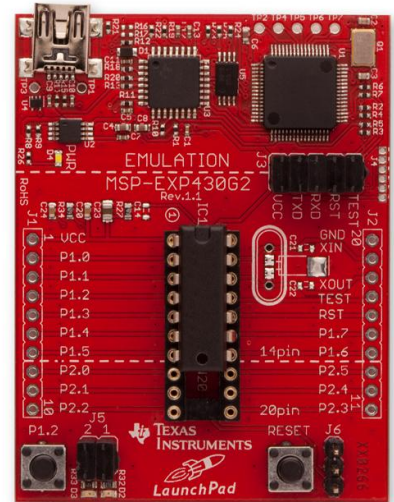
- O theremin: originalmente duas antenas metálicas e osciladores, responsáveis por controlar a tonalidade e amplitude sonoras
- Proposta:
 - sensores fotoelétricos no lugar de antenas metálicas
 - controle por variação luminosa, ao invés de variação de campo magnético
 - unidade microcontrolada programável no lugar de osciladores de tonalidade
 - controle de amplitude com um circuito amplificador controlado por tensão (amplificador de ganho variável)

MSP430G2 LaunchPad

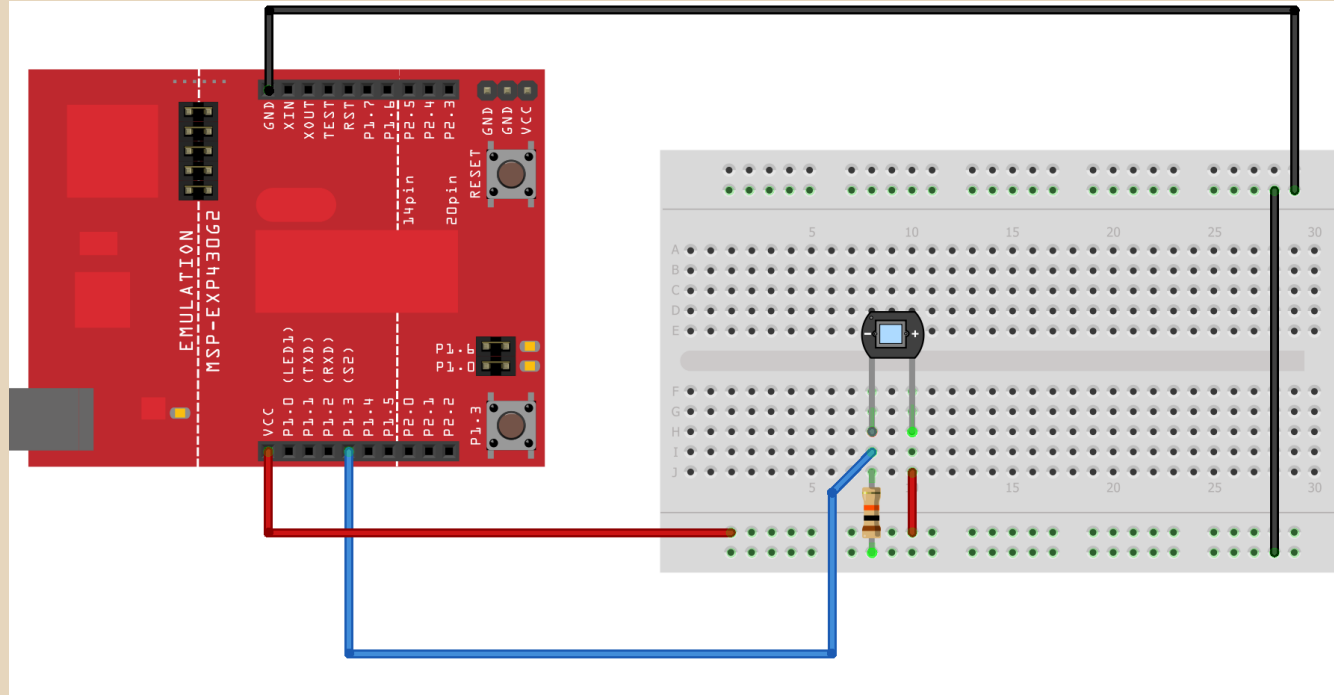
- Kit de desenvolvimento com plataforma open-source para prototipação de circuitos eletrônicos chamada Energia
 - compilador mspgcc, é baseada no Arduino
- Energia possui IDE baseada em Processing, o mesmo no qual Arduino se baseia
- Custo baixo: US\$ 9,99



Energia



Implementação - Controle de frequência



Implementação - Controle de frequência

- Programa desenvolvido - 3 etapas
 - Criação de variáveis
 - Setup
 - Loop

Implementação - Controle de frequência

```
1  /* Variáveis */
2  const int sensorFreq = A0; // seleciona pino de input para frequencia
3  const int freqPin = 15;    // saída de frequencia
4
5  int freqHigh = 0;
6  int freqLow = 999999;
7  int freqValue = 0;        // armazena valor capturado no sensor de frequencia
```

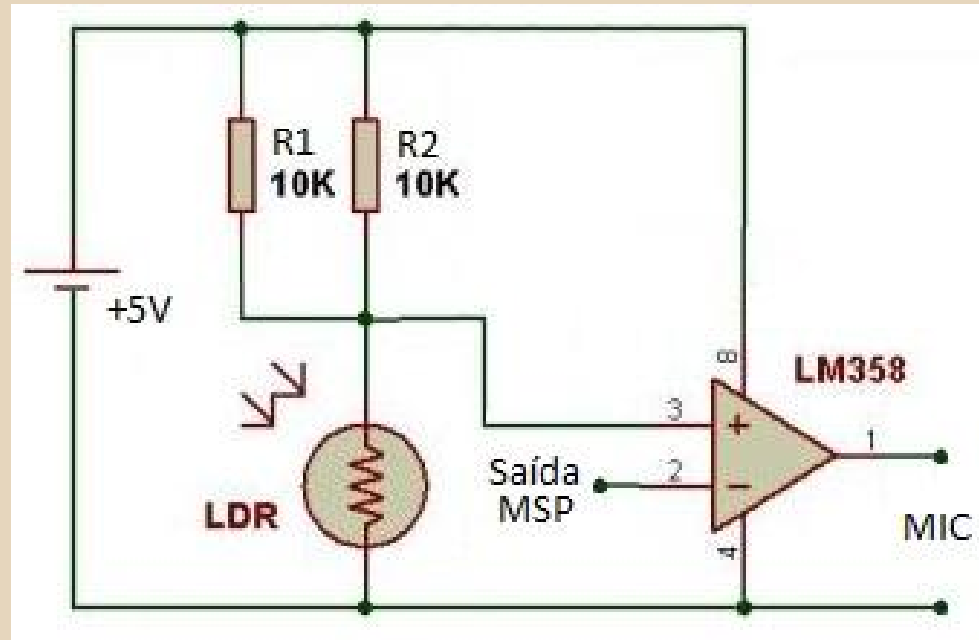
Implementação - Controle de frequência

```
9 void setup() {  
10     // seta pinos como input ou output  
11     pinMode(sensorFreq, INPUT);  
12     pinMode(freqPin, OUTPUT);  
13  
14     // Calibra o sensor de frequencia por 5 segundos  
15     while(millis() < 5000) {  
16         freqValue = analogRead(sensorFreq);  
17  
18         // Registra o maior valor do sensor  
19         if(freqValue > freqHigh) {  
20             freqHigh = freqValue;  
21         }  
22  
23         // Registra o menor valor do sensor  
24         if(freqValue < freqLow) {  
25             freqLow = freqValue;  
26         }  
27     }  
28 }
```

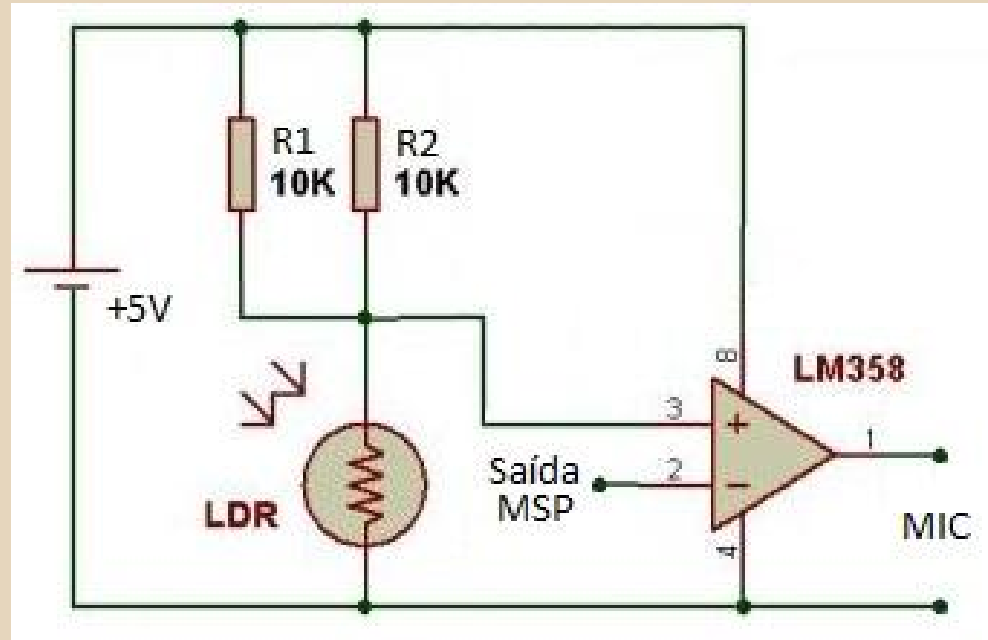
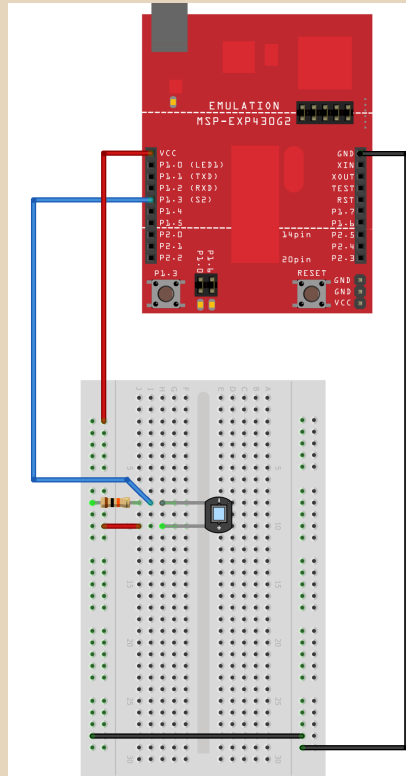
Implementação - Controle de frequência

```
31 void loop() {  
32     // Lê a frequência do pino A0  
33     freqValue = analogRead(sensorFreq);  
34  
35     //map the sensor values to a wide range of pitches  
36     //Adjust the values below to conform the maximum and  
37     //minimum numbers you get from the sensor  
38     int pitchFreq = map(freqValue, freqLow, freqHigh, 20, 1000);  
39  
40     //toca tom  
41     tone(freqPin, pitchFreq, 20);  
42  
43     //espera  
44     delay(10);  
45 }
```

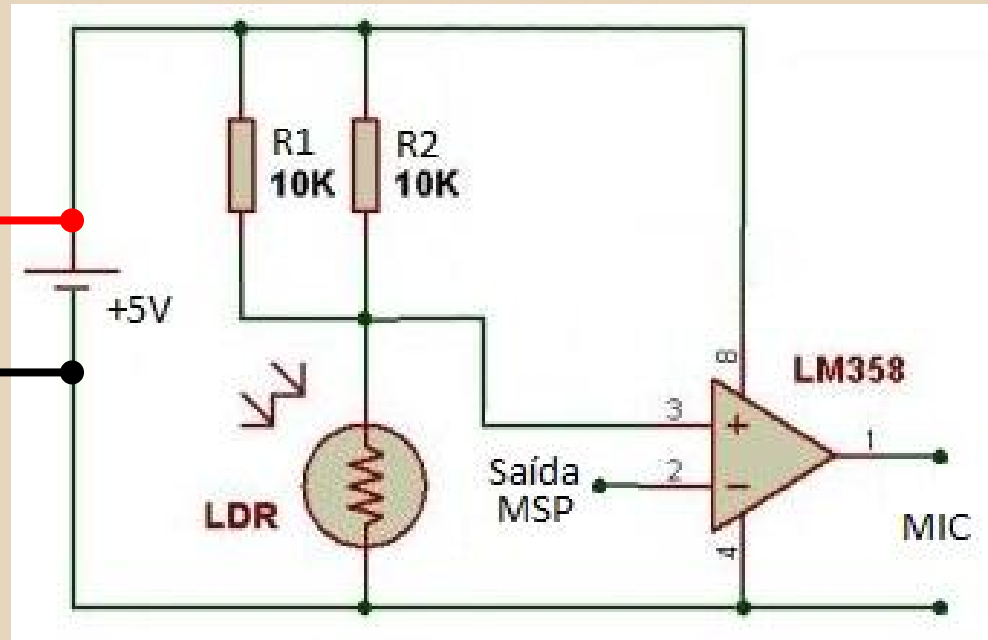
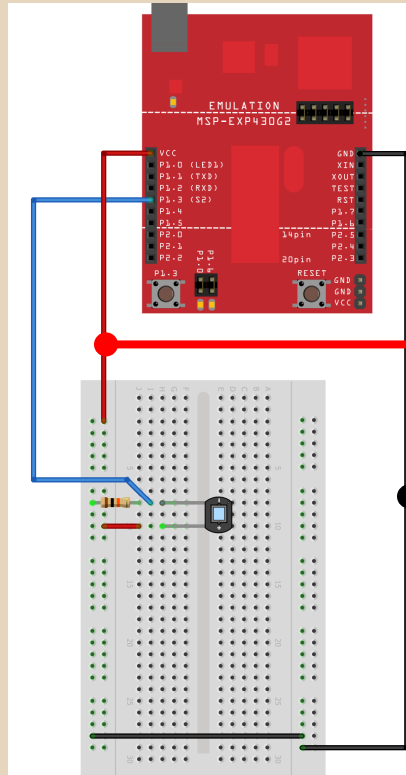
Implementação - Controle de amplitude



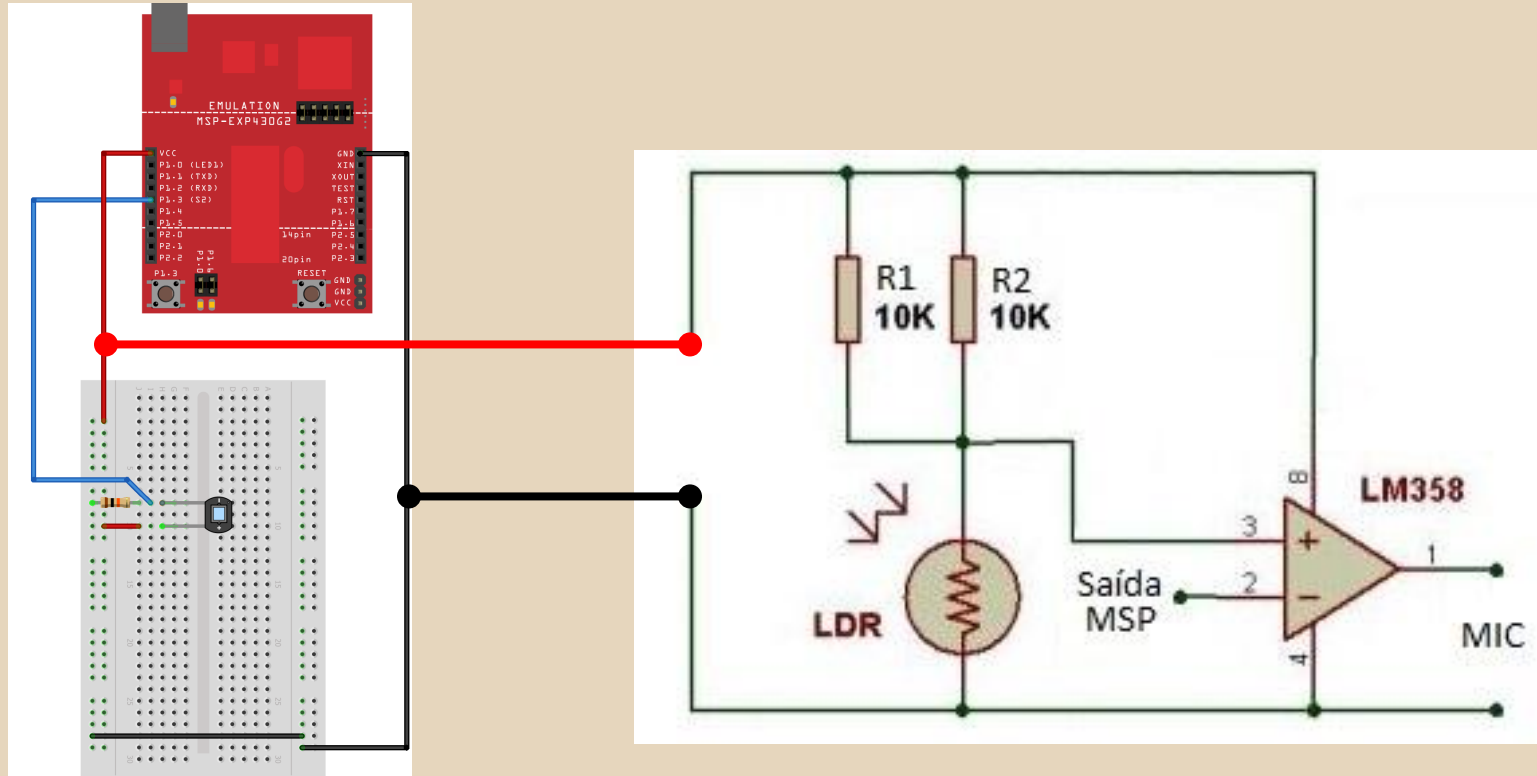
Implementação - Integração



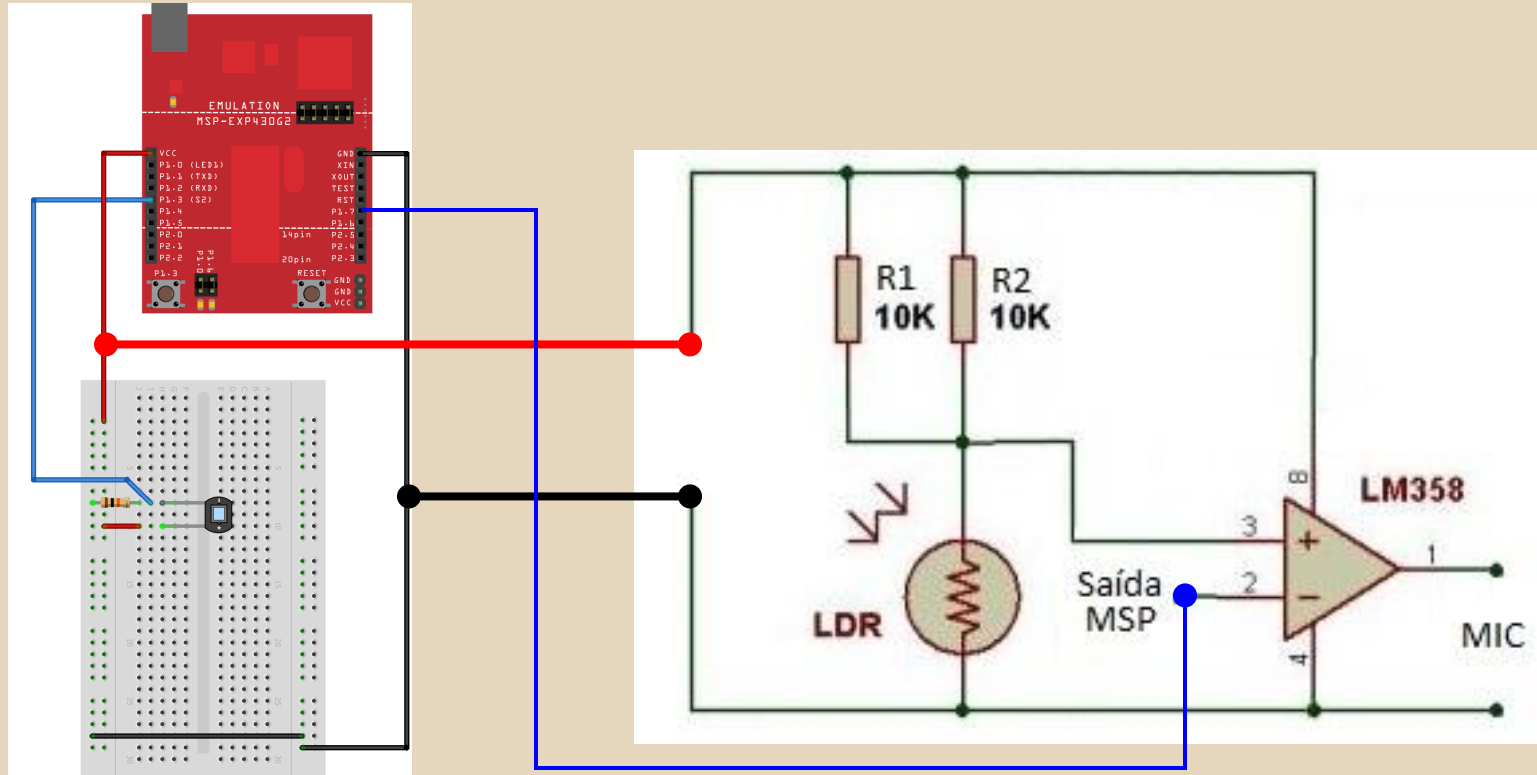
Implementação - Integração



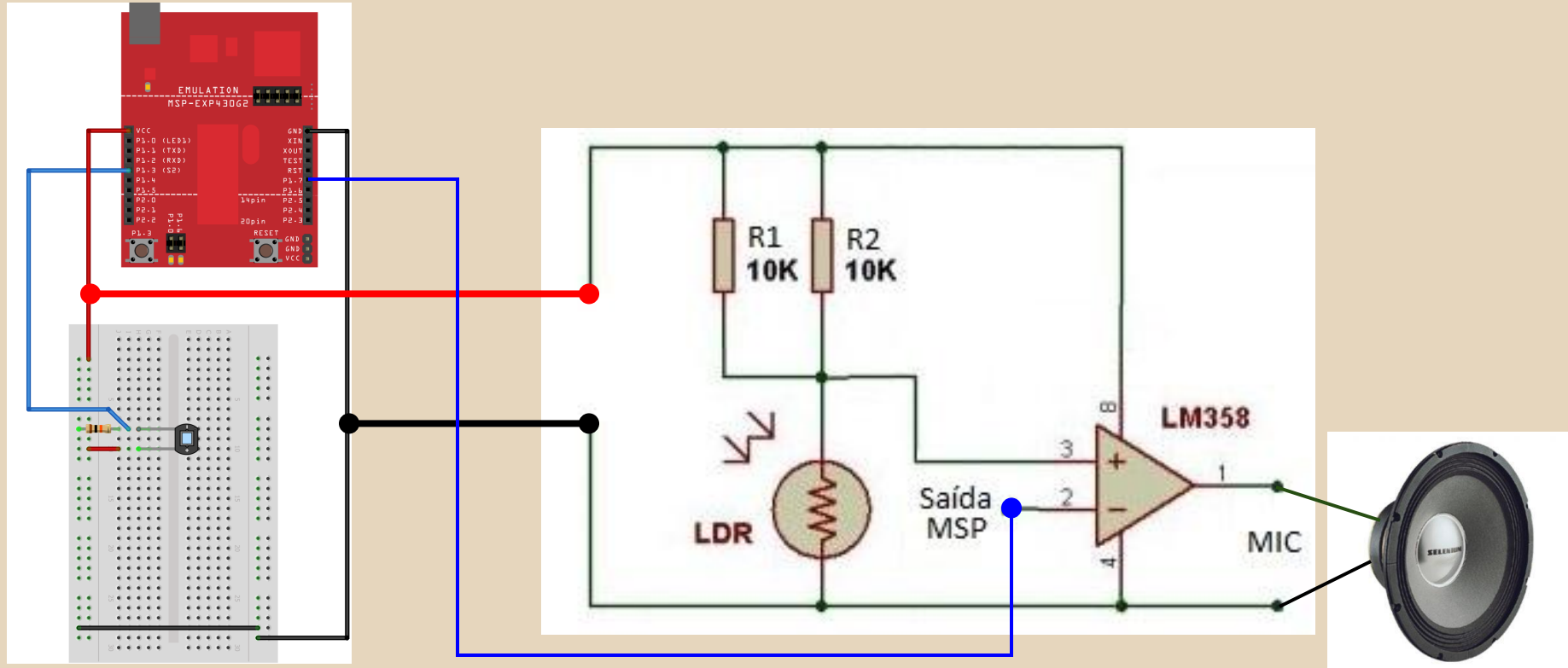
Implementação - Integração



Implementação - Integração



Implementação - Integração



Demonstração

Resultados Obtidos

- A responsividade dos sensores fotoelétricos foi satisfatória
- Sensor para controle de frequência
 - teve uma responsividade muito boa
 - capturando um bom espectro de frequências
 - facilidade em destoar tons reproduzidos pelo alto-falante a medida que se aproxima ou se afasta a mão do sensor
- Sensor para controle de amplitude
 - sensibilidade muito alta
 - maximizando ou corta o som por definitivo

Considerações Finais

- Oportunidade de trabalhar com hardwares que até então não tive contato no curso de ciência da computação.
- Possibilidade de poder simular um instrumento musical dos anos 30
 - Necessidade de apenas alguns componentes de hardware e pequenos trechos de código
 - Versatilidade das tecnologias que hoje temos à disposição
 - Facilidade de acesso a essas tecnologias
- Custo de projeto baixo: MSP430 + Ampop + resistores + sensores + alto-falante = R\$ 30,00

Obrigado.

Perguntas?