



HARP LASER MUSIC

Developed by:
Kreisler Brenner Mendes



1-INTRODUÇÃO

2- Requisitos;

3- Arquitetura;

4- Materiais;

5- Montagem;

6- Circuito

7- Código



2 - REQUISITOS

Funcionais

- <RF001>. Deve ser capaz emitir o som das notas da escala natural de DÓ maior;
- <RF002>. Deve permitir que usuário escolha uma música para ser tocada e iniciar o jogo;
- <RF003>. Deve ser capaz de comparar as notas tocadas pelo usuário com a música escolhida pelo mesmo;
- <RF004>. Deve ser capaz calcular uma pontuação que qualifique a habilidade musical do usuário;



2 - REQUISITOS

Não-Funcionais

- <RNF001>. Deve ser capaz de mostrar visualmente as notas a serem tocadas da música escolhida.



3 - ARQUITETURA

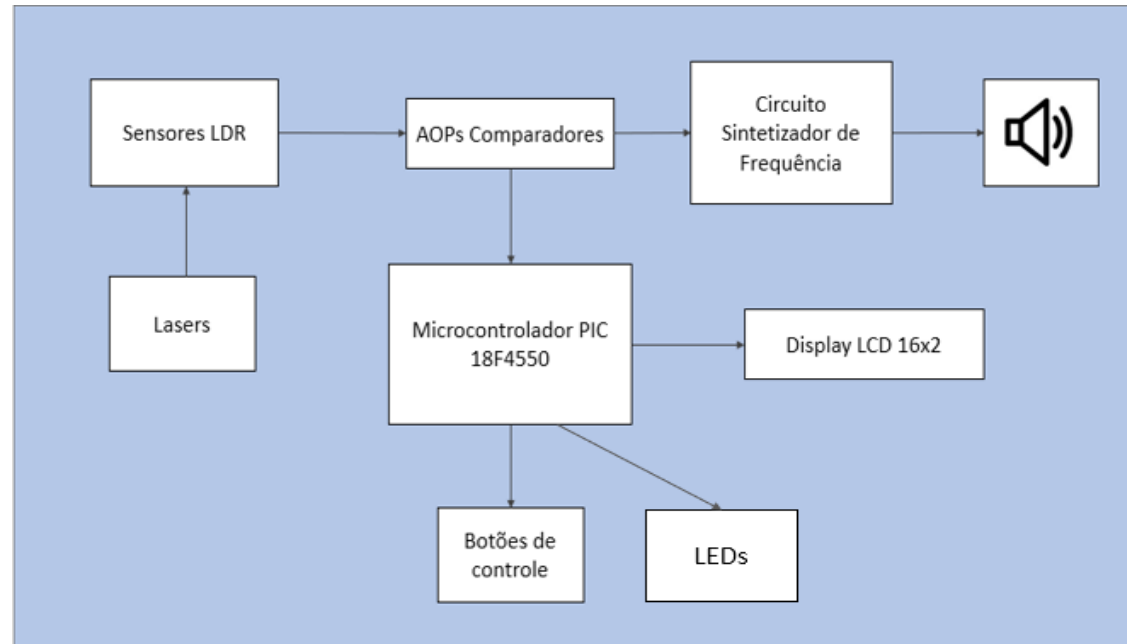
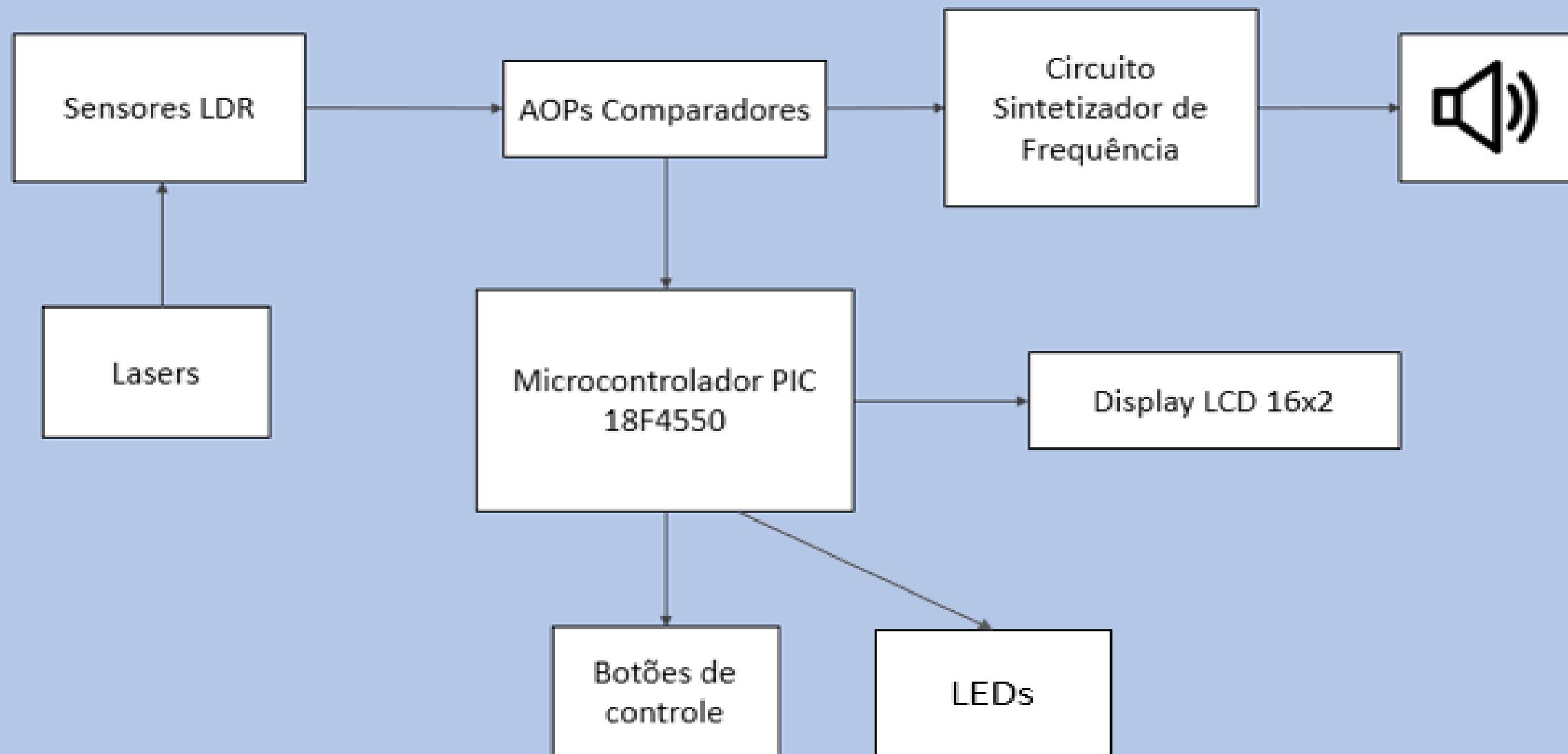


Fig.1 Arquitetura global do sistema



4 - MATERIAIS



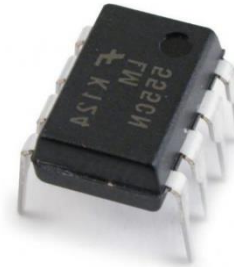
1x Microcontrolador PIC
18F4550



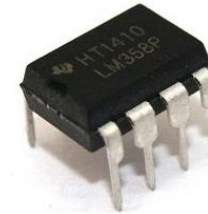
8x Laser Mini 650nm
6mm 5V 5mW



8x LDR



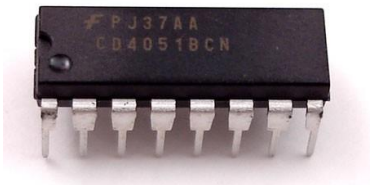
1x LM555



4x LM358 (Comparador)



4x Botões



1x CD4051 (Mux)



3x SN7408 (AND)



8x Trimpots 100kΩ



1x Display LCD16x2



8x Leds Amarelos



1x Alto
Falante

5 - MONTAGEM (Protótipo)

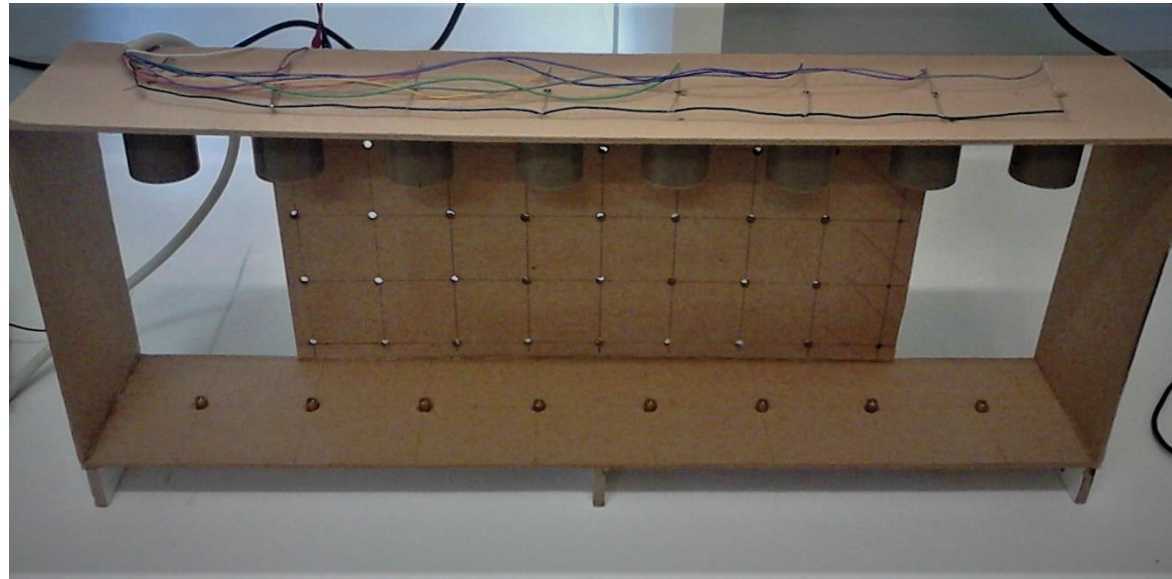
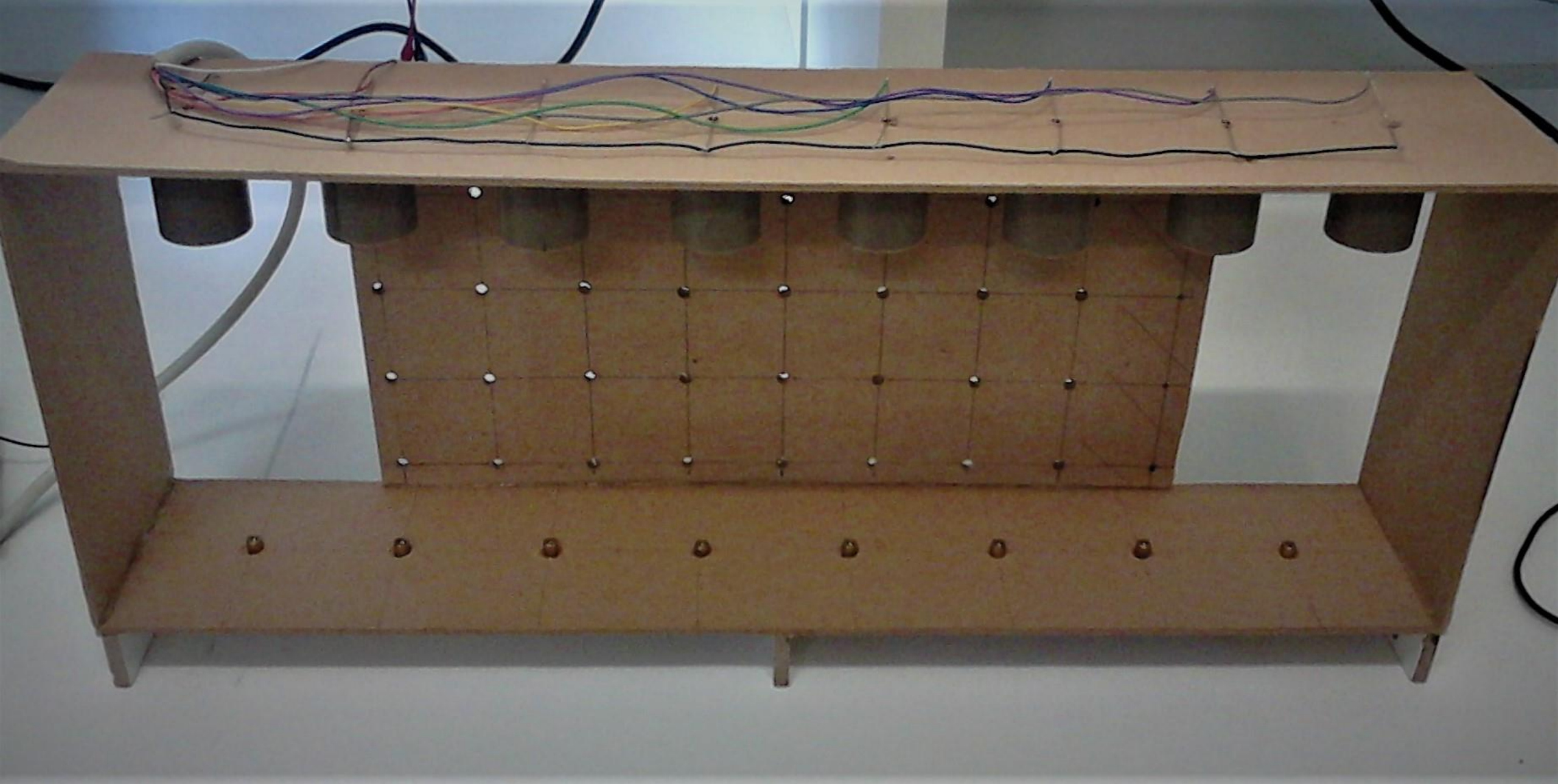
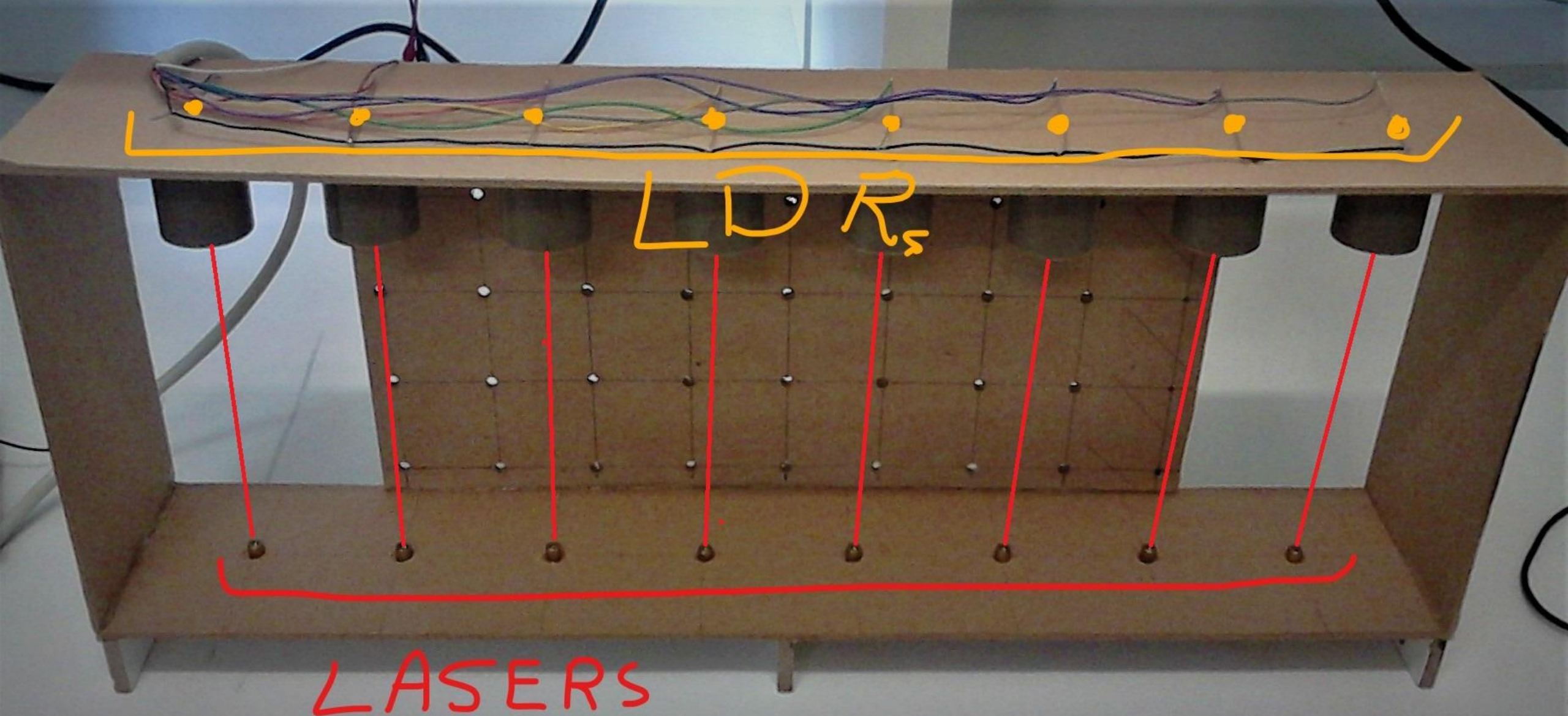
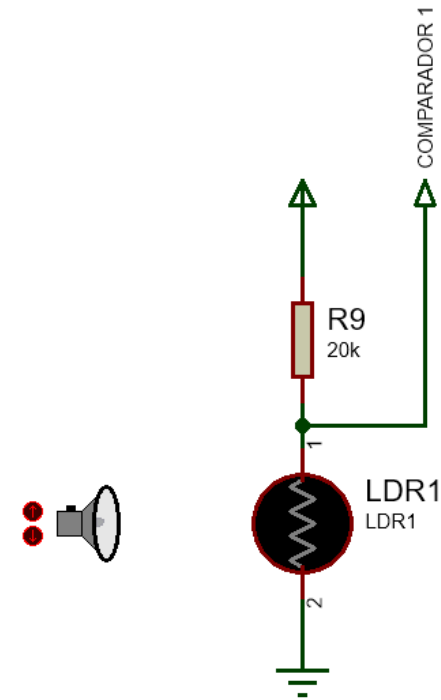
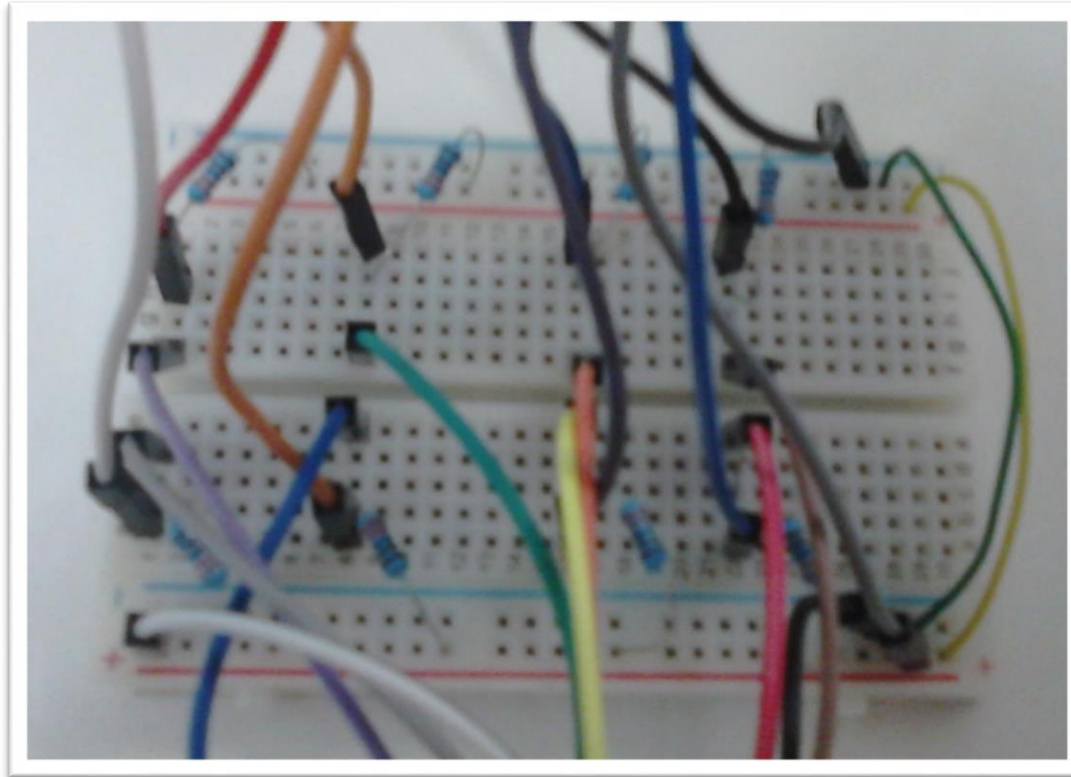


Fig.1 – Estrutura em mdf com lasers fixados em sua base igualmente espaçado e cada um apontado para um respectivo sensor de luz.

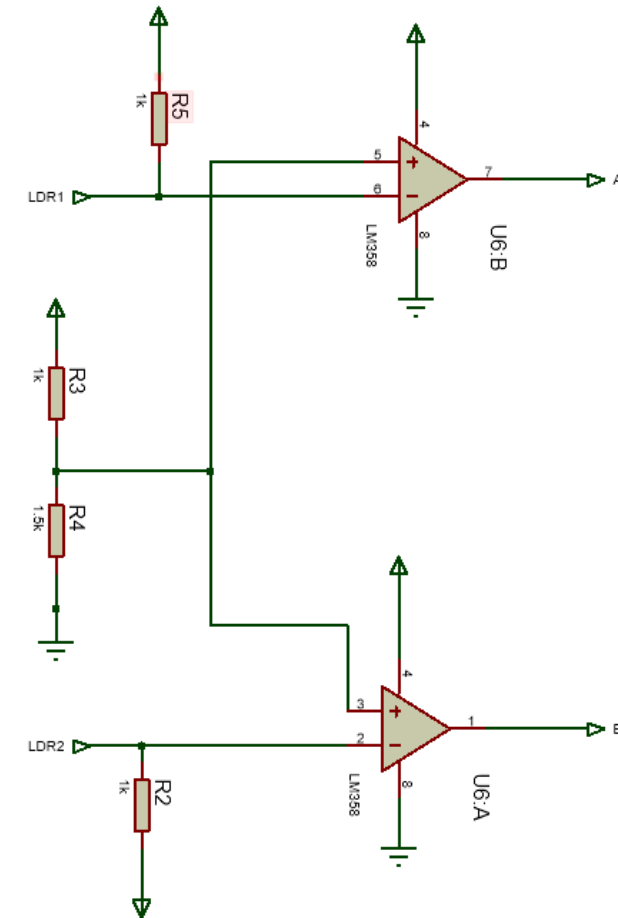
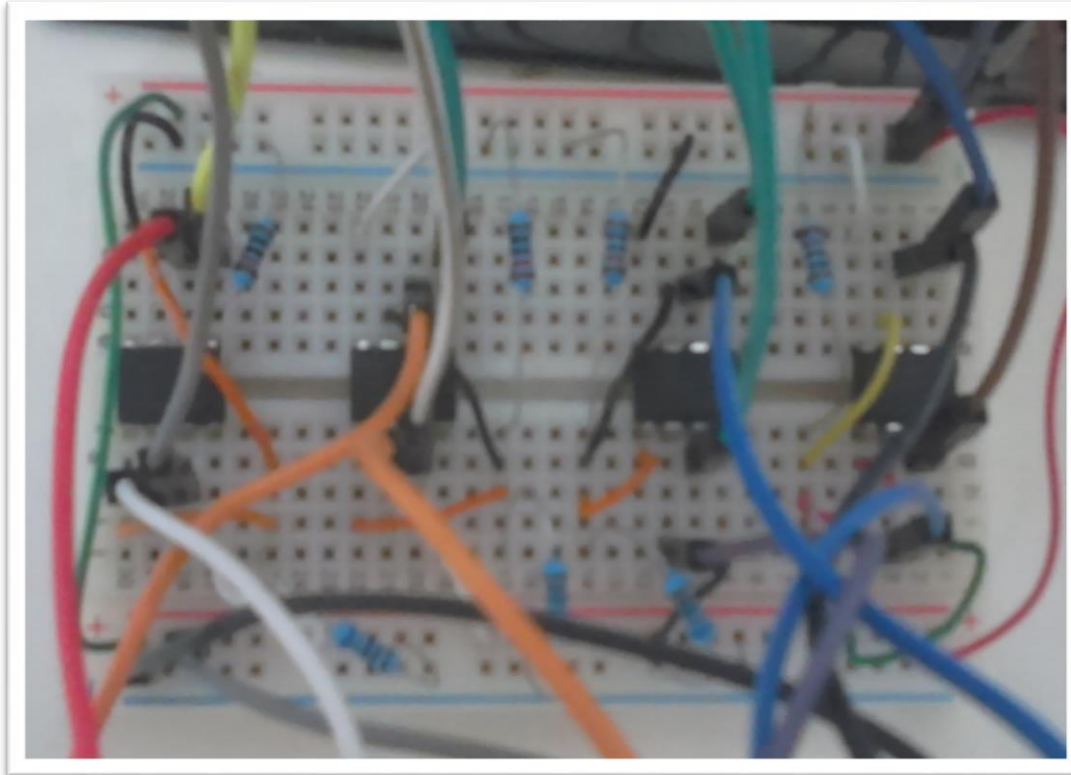




6.1 - CIRCUITO (Polarização LDR)



6.2 - CIRCUITO (Comparadores)



6.3 CIRCUITO (Circuito lógico de um Encoder)

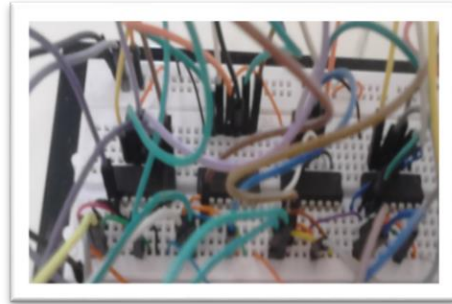
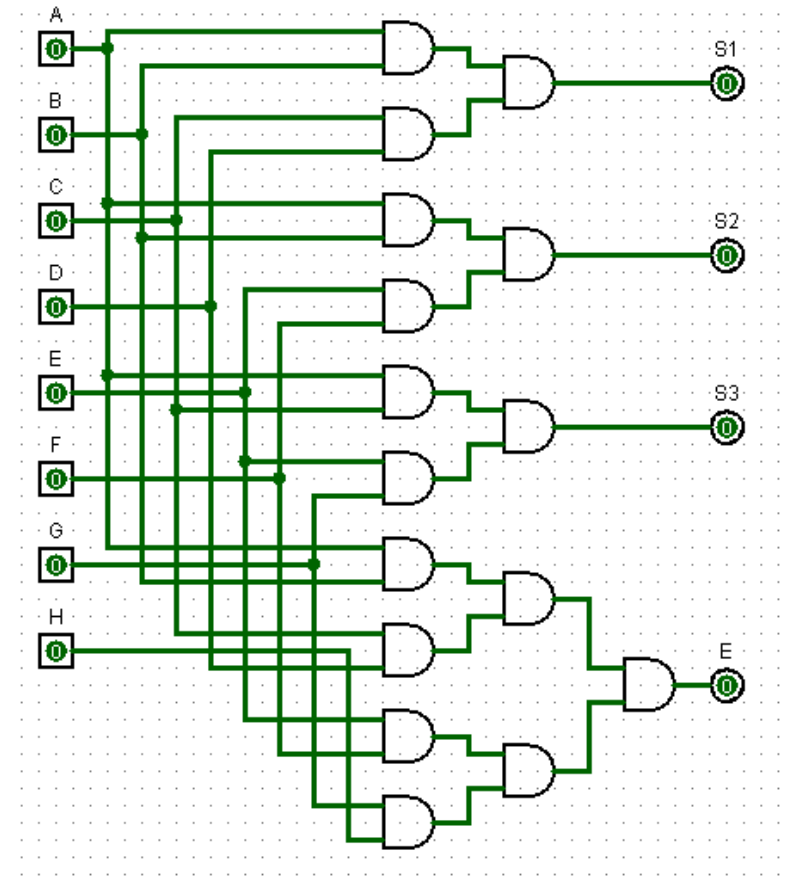
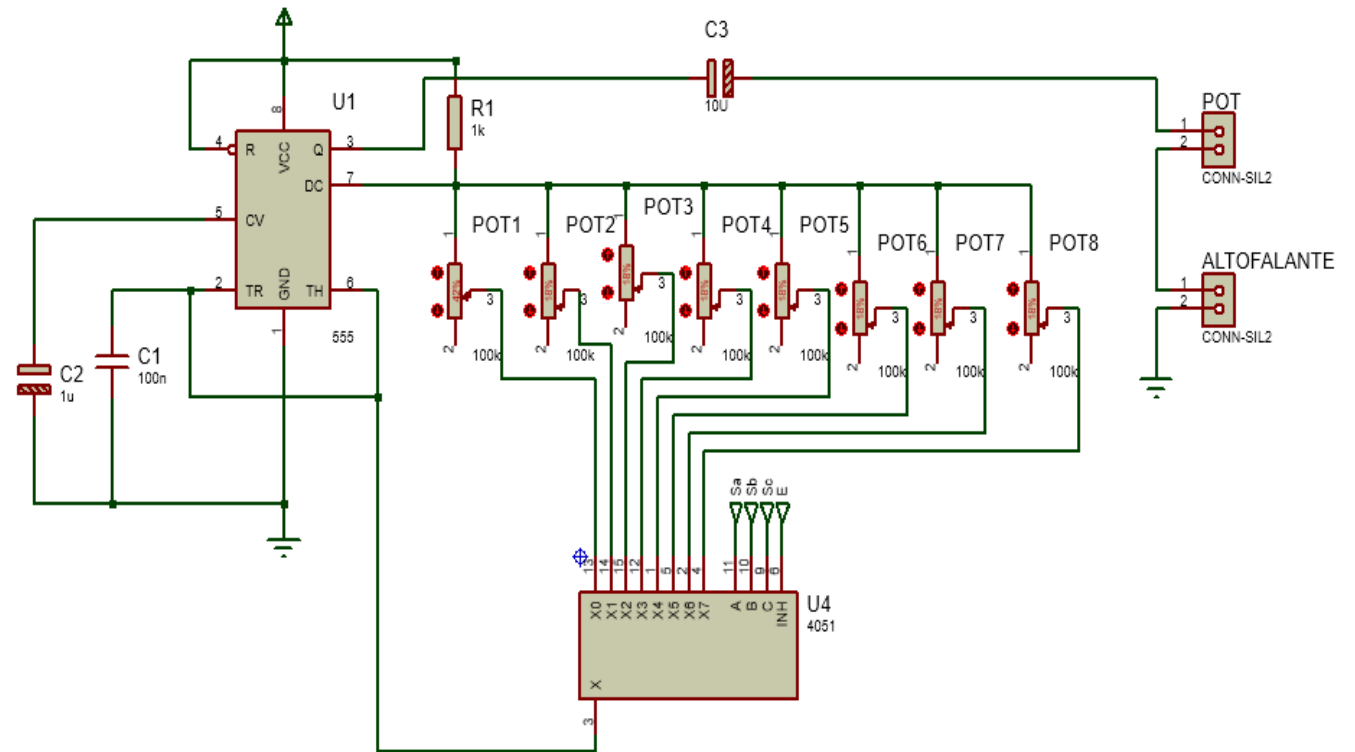
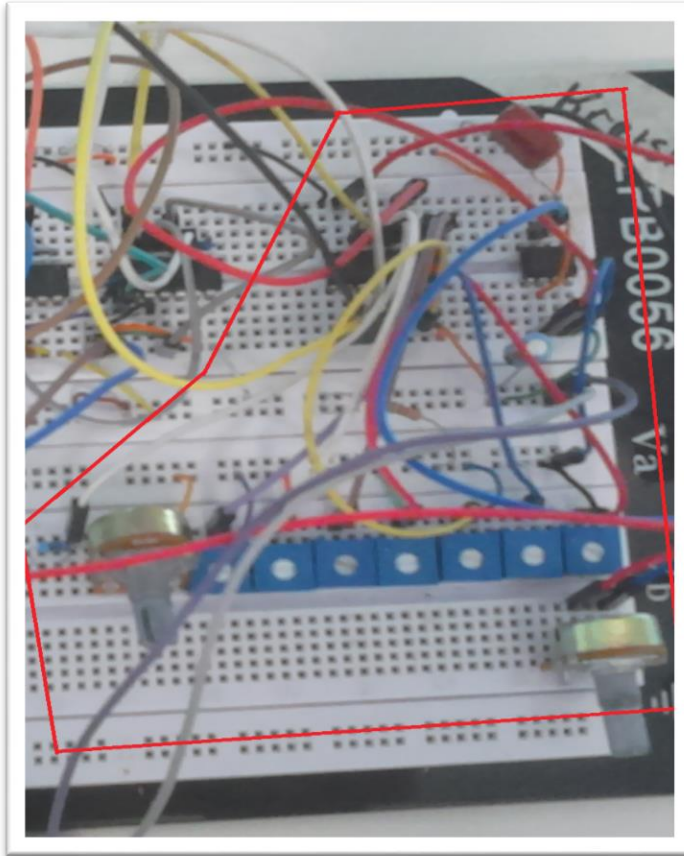


TABELA VERDADE

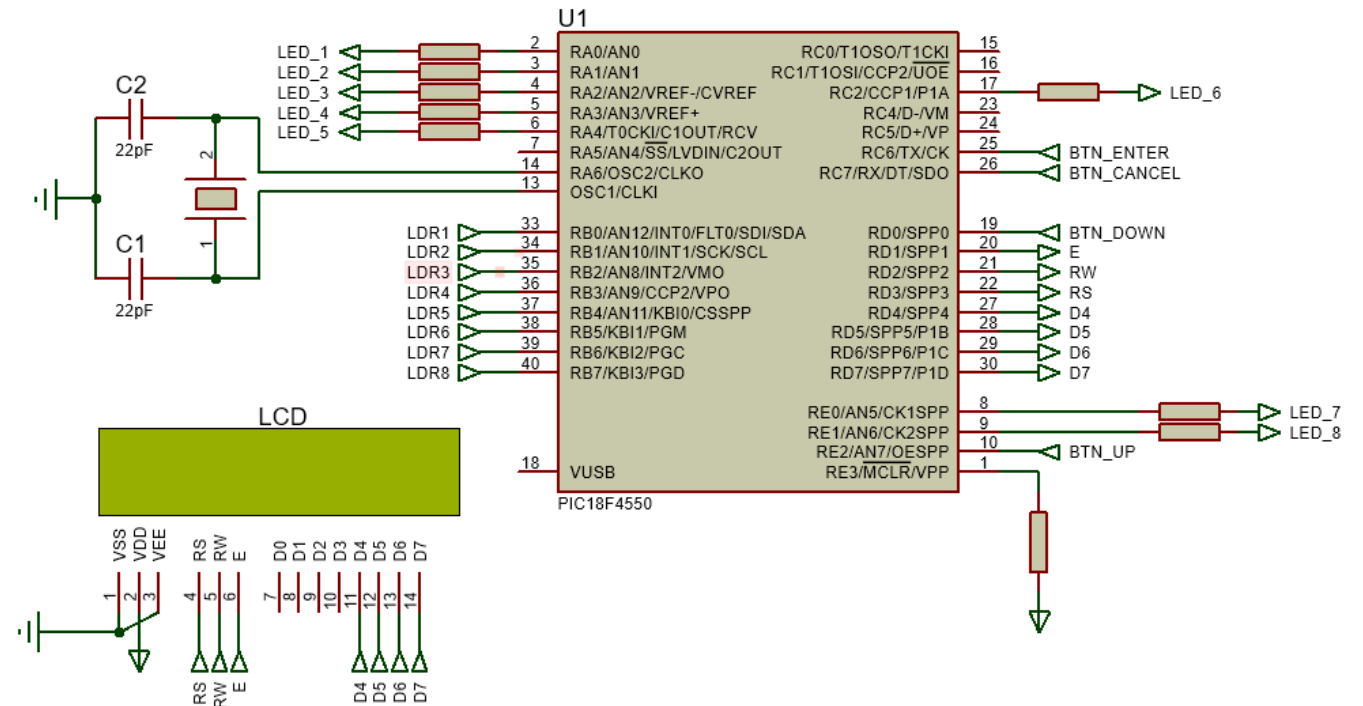
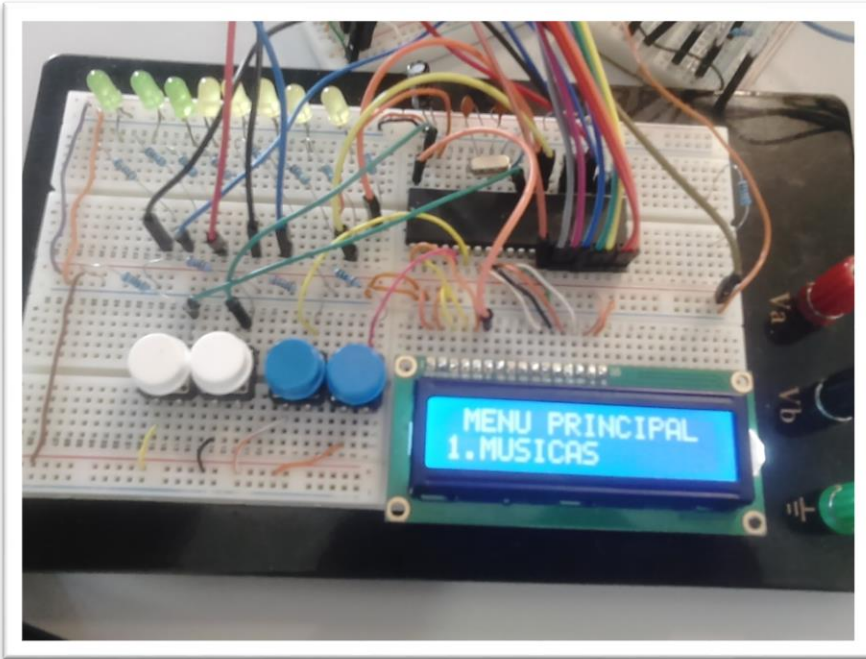
A	B	C	D	E	F	G	H	S1	S2	S3	E
1	1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	1	0	1	0	0	1	0
1	1	1	1	1	0	1	1	0	1	0	0
1	1	1	1	0	1	1	1	0	1	1	0
1	1	1	0	1	1	1	1	1	0	0	0
1	1	0	1	1	1	1	1	1	0	1	0
1	0	1	1	1	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	1	1	1	0



6.4 - CIRCUITO (Sintetizador de Frequência)



6.5 - CIRCUITO (Interface com microcontrolador)



7 - Código (Header)

```
50 const int8 DO_3 = 0b11111110;
51 const int8 RE_3 = 0b11111101;
52 const int8 MI_3 = 0b11111011;
53 const int8 FA_3 = 0b11110111;
54 const int8 SOL_3 = 0b11101111;
55 const int8 LA_3 = 0b11011111;
56 const int8 SI_3 = 0b10111111;
57 const int8 DO_4 = 0b01111111;
58 const int8 PAUSA = 0b11111111;
59
60 CONST int QUANT_MUSICAS = 4;
61
62
63 int menuPrincipal(); // Menu
64 int listaMusicas(); // Menu 1
65 void records(); // Menu 2 - TOP 10 das melhores pontuações
66
67 void carregar(); // Prepara o usuário para começar a jogar
68 void notesToLeds(int8 note); //Acende o led que corresponde a nota a ser tocada
69 int tocar_musica(int index); // Retorna a pontuação dada em percentual
70 void qualificarPrecisao(int8 index,int8 acertos_nota);//Mostra ao usuário o percentual
71
72 /*Compara a pontuação obtida pelo usuario com as armazenadas na memória (TOP 10), caso
73 a pontuação seja igual ou maior à pontuação comparada, então substitui-se a obtida pela
74 comparada e desloca uma posição abaixo todas as outras pontuações inferiores.
75 */
76 void salvar(int8 pontuacao,int8 index);
77
78
79 struct musica {
80 char titulo[16];
81 int tempo;
82 };
83
84 struct musica sound[QUANT_MUSICAS];
85
```

```
86 //VARIÁVEIS PARA CONTROLE DO MENU ROTATIVO
87 int index_Nivel1 = 0;
88 int index_Nivel2 = 0;
89 boolean flag_control = 1;
90 boolean flag_ENTER = FALSE;
91 boolean flag_CLOSE = FALSE;
92 boolean flag_DOWN = FALSE;
93 boolean flag_UP = FALSE;
```




```

50 const int8 DO_3 = 0b11111110;
51 const int8 RE_3 = 0b11111101;
52 const int8 MI_3 = 0b11111011;
53 const int8 FA_3 = 0b11110111;
54 const int8 SOL_3 = 0b11101111;
55 const int8 LA_3 = 0b11011111;
56 const int8 SI_3 = 0b10111111;
57 const int8 DO_4 = 0b01111111;
58 const int8 PAUSA = 0b11111111;
59
60 CONST int QUANT_MUSICAS = 4;
61
62
63 int menuPrincipal(); // Menu
64 int listaMusicas(); // Menu 1
65 void records(); // Menu 2 - TOP 10 das melhores pontuações
66
67 void carregar(); // Prepara o usuário para começar a jogar
68 void notesToLeds(int8 note); //Acende o led que corresponde a nota a ser tocada
69 int tocar_musica(int index); // Retorna a pontuação dada em percentual
70 void qualificarPrecisao(int8 index,int8 acertos_nota);//Mostra ao usuário o percentual
71
72 /*Compara a pontuação obtida pelo usuario com as armazenadas na memória (TOP 10), caso
73 a pontuação seja igual ou maior à pontuação comparada, então substitui-se a obtida pela
74 comparada e desloca uma posição abaixo todas as outras pontuações inferiores.
75 */
76 void salvar(int8 pontuacao,int8 index);
77
78
79 struct musica {
80 char titulo[16];
81 int tempo;
82 };
83
84 struct musica sound[QUANT_MUSICAS];

```

```

86 //VARIAVEIS PARA CONTROLE DO MENU ROTATIVO
87 int index_Nivel1 = 0;
88 int index_Nivel2 = 0;
89 boolean flag_control = 1;
90 boolean flag_ENTER = FALSE;
91 boolean flag_CLOSE = FALSE;
92 boolean flag_DOWN = FALSE;
93 boolean flag_UP = FALSE;

```