

Projeto 11; Melodia

(Beginning Arduino)

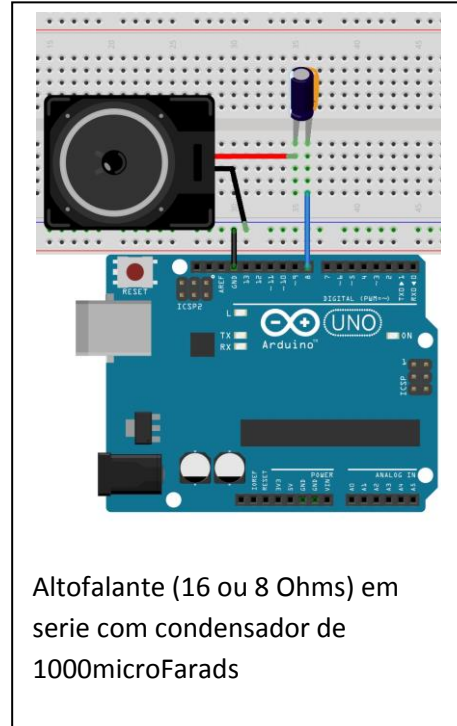
Copyright © 2010 by Michael McRoberts)

Este programa gera uma melodia.

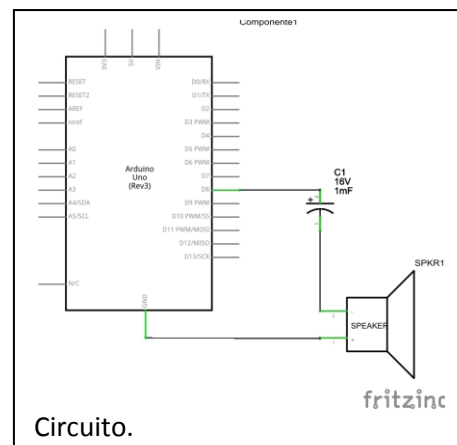
// Puff the magic Dragon lived by the sea.

```
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
#define WHOLE 1
#define HALF 0.5
#define QUARTER 0.25
#define EIGHTH 0.125
#define SIXTEENTH 0.0625
int tune[] = { NOTE_C4, NOTE_C4, NOTE_C4, NOTE_C4, NOTE_C4, NOTE_B3, NOTE_G3, NOTE_A3,
· NOTE_C4, NOTE_C4, NOTE_G3, NOTE_G3, NOTE_F3, NOTE_F3, NOTE_G3, NOTE_F3, NOTE_E3,
NOTE_G3,· NOTE_C4, NOTE_C4, NOTE_C4, NOTE_C4, NOTE_A3, NOTE_B3, NOTE_C4,
NOTE_D4};
float duration[] = { EIGHTH, QUARTER+EIGHTH, SIXTEENTH, QUARTER, QUARTER, HALF, HALF,
· HALF, QUARTER, QUARTER, HALF+QUARTER, QUARTER, QUARTER, QUARTER,
QUARTER+EIGHTH, EIGHTH,· QUARTER, QUARTER, QUARTER, EIGHTH, EIGHTH, QUARTER,
QUARTER, QUARTER, QUARTER,· HALF+QUARTER};
int length;
void setup() {
    pinMode(8, OUTPUT);
    length = sizeof(tune) / sizeof(tune[0]);
}
void loop() {
    for (int x=0; x<length; x++) {
        tone(8, tune[x]);
        delay(1500 * duration[x]);
        noTone(8);
    }
    delay(5000);
}

#define
define um valor. For example:
```



Altofalante (16 ou 8 Ohms) em
serie com condensador de
1000microFarads



```
#define display_Altura 8
#define display_largura 32
```

No programa, definimos as notas. Na melodia a primeira nota é C4, frequência 262Hz. O Dó médio (centro do piano) da escala musical.

As outras 5 **#define** são para os comprimentos das notas.

A duração em segundos de uma nota depende do compasso: o tempo da nota em uma música pode ser diferente em outra música, se o compasso for diferente. Dessa forma, a representação da duração sugere uma fração do tempo do compasso.

Então temos as notas semibreve (um tempo), mínima (meio tempo), semínima (um quarto de tempo), colcheia (um oitavo de tempo), semicolcheia (1/16 de tempo), fusa (1/32 de tempo) e semifusa (1/64 de tempo).

Por exemplo, um quarto de uma nota (semiminima) é 0,25. Multiplicamos por 1500ms que é a nota inteira (semibreve).
= 375 milliseconds

A função **sizeof(parametro)**

Retorna o numero de bytes do parametro passado a ela . Exemplo:

```
sizeof(tune)=26*2
```

```
sizeof(tune[0])=2
```

Portanto **length= sizeof(tune) / sizeof(tune[0]);**

Retorna o valor 26. A musica tem 26 notas.

Agora, é só acompanhar o programa, entender, e escrever as melodias que gostar.

