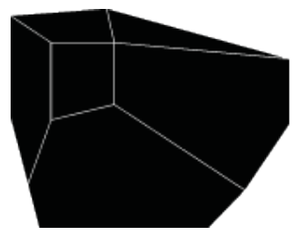
Arduino Synth

Sintetizadores DIY



casa da música

DIGITÓPIA

Digitópia

A Digitópia é uma plataforma de música digital sediada na Casa da Música, no Porto, que incentiva a audição, a performance e a criação musical. Baseando-se em ferramentas digitais, embora não exclusivamente, enfatiza a criação musical colaborativa, o design de software, a educação musical e a inclusão social, promovendo a emergência de comunidades multiculturais de performers, compositores, curiosos e amantes de música. Com uma presença física composta por 12 computadores e diversos controladores musicais, recebe alunos de várias escolas, promove seminários, é visitada por curiosos, turistas, crianças e seniores, proporcionando experiências musicais auditivas e criativas, enriquecedoras para todos aqueles que as vivenciam.

www.casadamusica.com/digitopia
www.facebook.com/DigitopiaCasaDaMusica
www.facebook.com/groups/digitopiacasadamusica
www.github.com/Digitopia

casa da música
DIGITÓPIA

Digitópia Collective

Singular no panorama nacional, o Digitópia Collective é constituído pelos formadores do Serviço Educativo associados à Digitópia, a plataforma artística da Casa da Música reservada à criação musical em suporte tecnológico.
No seu trabalho, o ensemble aplica processos e modelos tão diversos quanto o design de instrumentos digitais, a concepção de hardware próprio, o circuit-bending, a exploração das relações entre imagem e som, a prática de VJ's e DJ's, a digital media ou os sistemas digitais interactivos. Da confluência de linguagens, trazidas por cada elemento do grupo, surge um repertório de música electrónica e digital com um declarado carácter performativo.

Tiago Ângelo

Académico

Conservatório em Trombone, CMC

Música Electrónica e Produção Musical na ESART - IPCB

Mestrado Multimédia, Perfil de Sound Design e Música Interactiva na FEUP - UP

Profissional

Freelancer

Digitópia - Casa da Música

Composição e media interactivos para artes performativas (DEMO, Radar360)

Artístico

Composição e performance Electrónica

Instalações interactivas

Construção de instrumentos musicais e desenvolvimento de software musical

Tiago Ângelo
tiagoangelo.tk

casa da música
DIGITÓPIA

Plano (sábado)

Manhã

(Breve) Introdução ao Arduino

Introdução à biblioteca Mozzi

Tarde

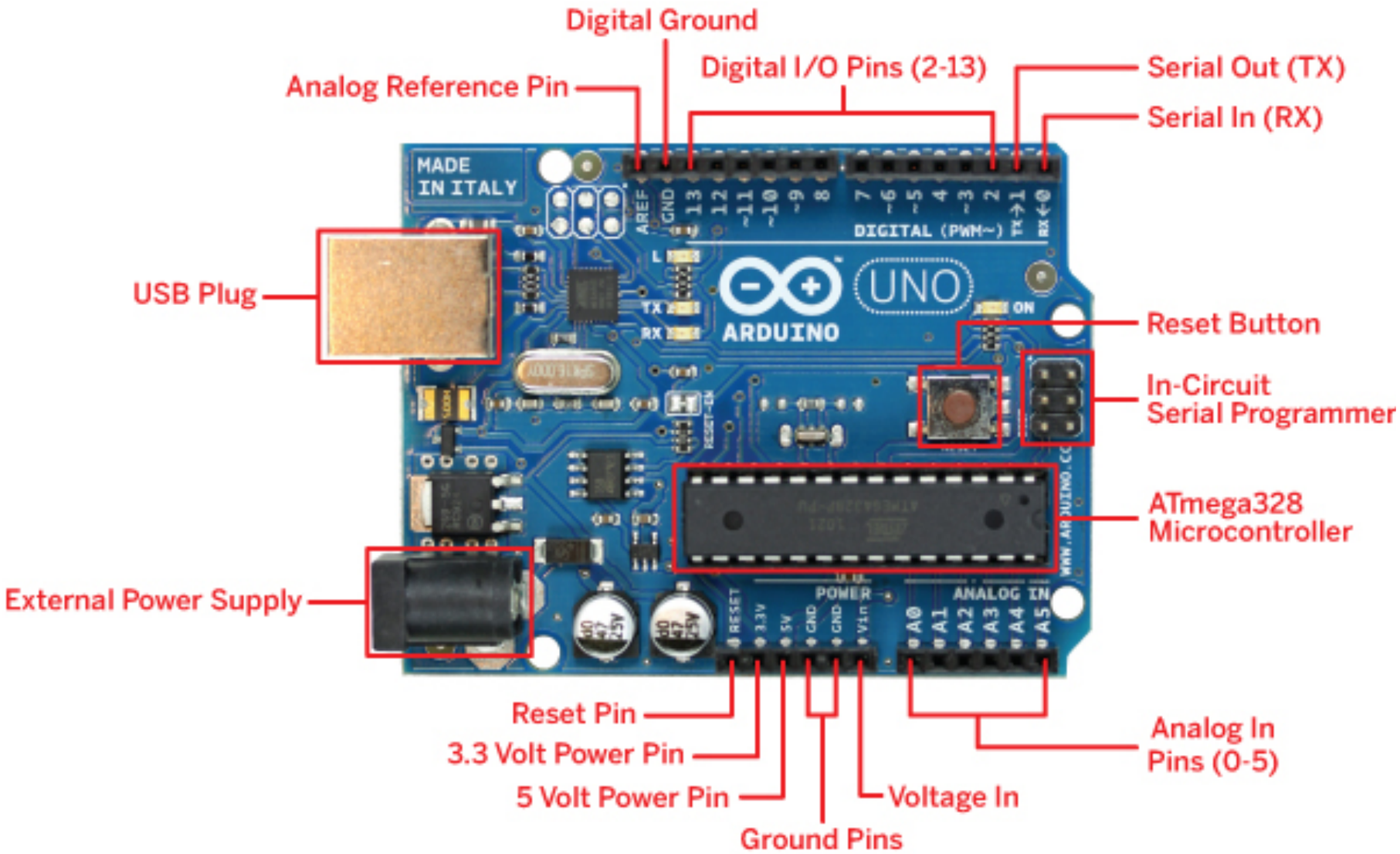
Hands On - construção de processadores de sinal

http://arduino.cc/

retirado de http://makezine.com/2015/03/19/massimo-banzi-fighting-for-arduino/

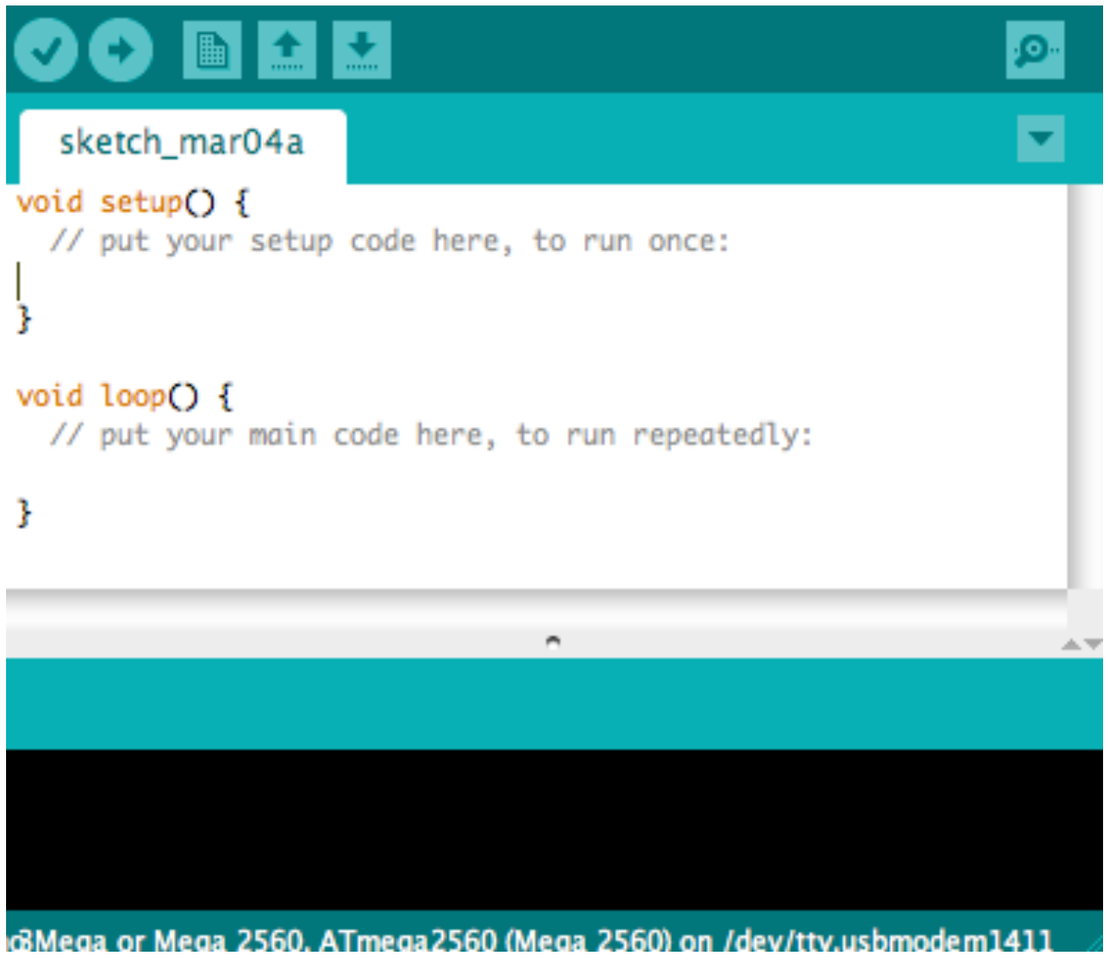
- Plataforma de prototipagem electrónica *open source*;
- Desenvolvida por Maximo Banzi, Tom Igoe, David Cuartielles, David Mellis e Gianluca Martino que começou por ser desenvolvida em 2005 em Ivrea, Itália;
- Documentário: https://vimeo.com/18539129

Hardware



retirado de https://myp-tech.wikispaces.com/+The+Ardunio+Project

Software (IDE)

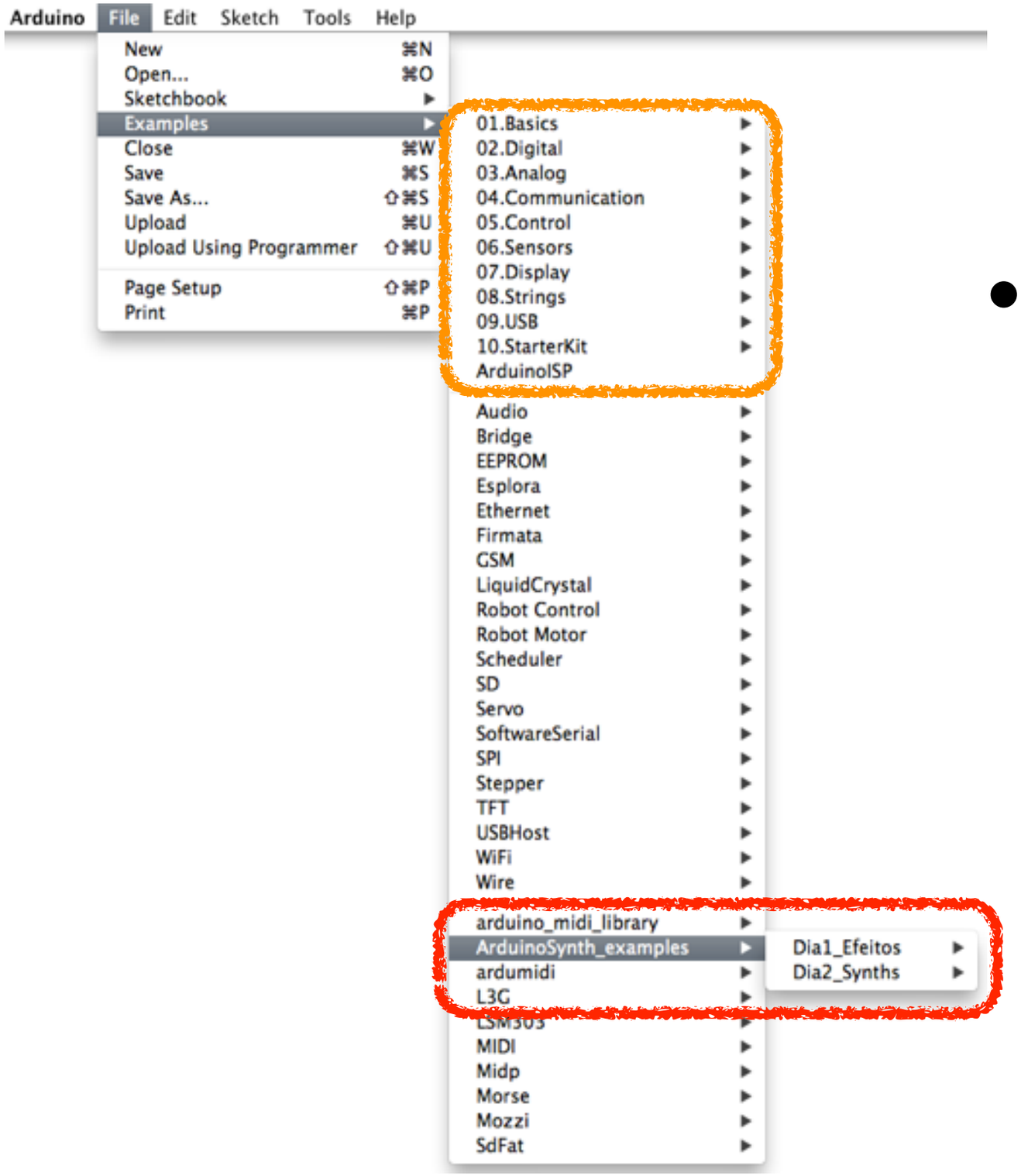


- IDE - *Integrated Development Environment*
- permite editar e carregar um programa para o microcontrolador
- semelhante ao Processing
- baseado em C++

Instalação

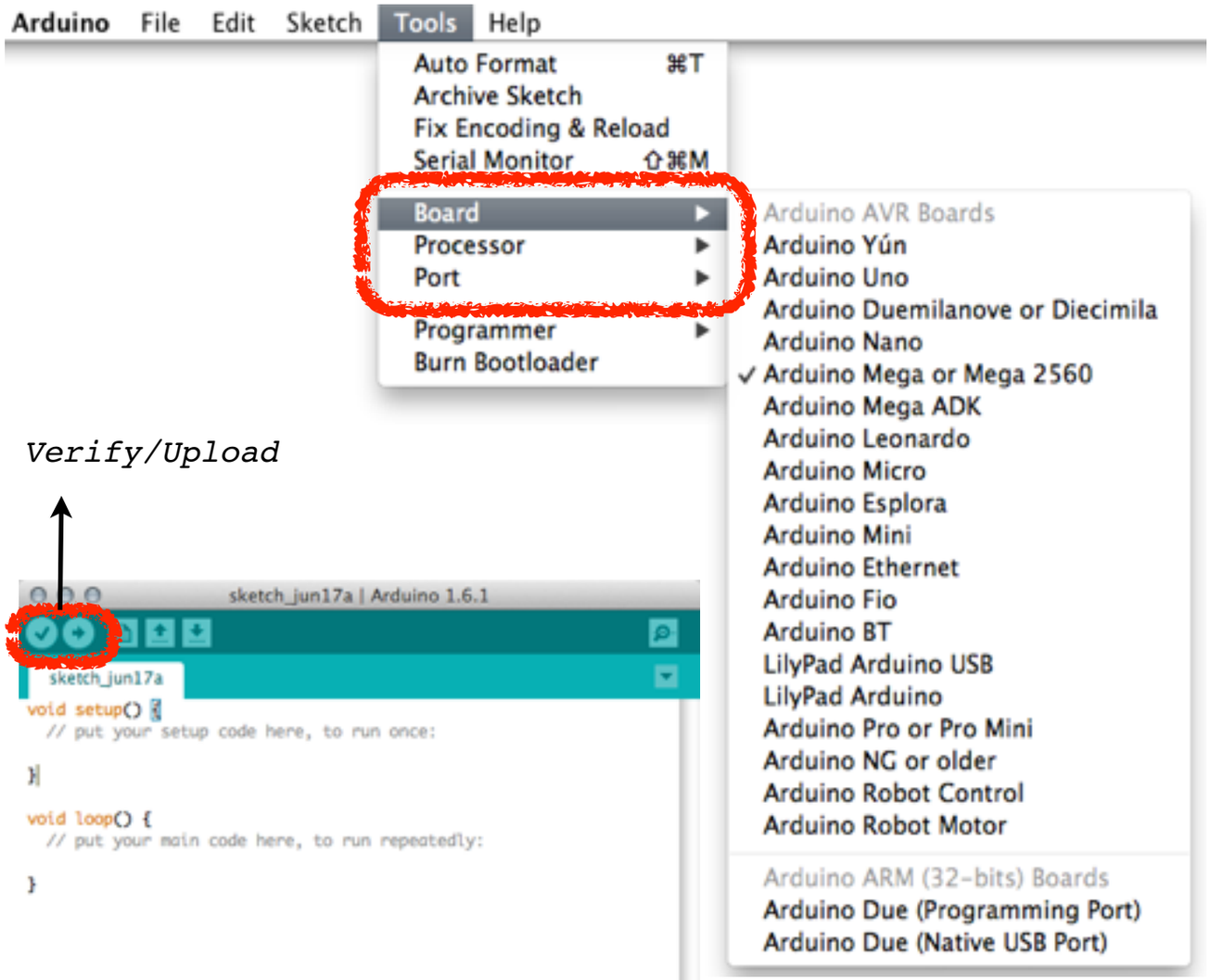
- **OSX :** http://arduino.cc/en/Guide/Howto
- **Windows :** http://arduino.cc/en/guide/windows
- Drivers :
- **Guia de instalação :** http://www.ftdichip.com/Support/Documents/InstallGuides.htm
- **Link para download :** http://www.ftdichip.com/Drivers/VCP.htm

Arduino - Iniciação



- Exemplos Arduino

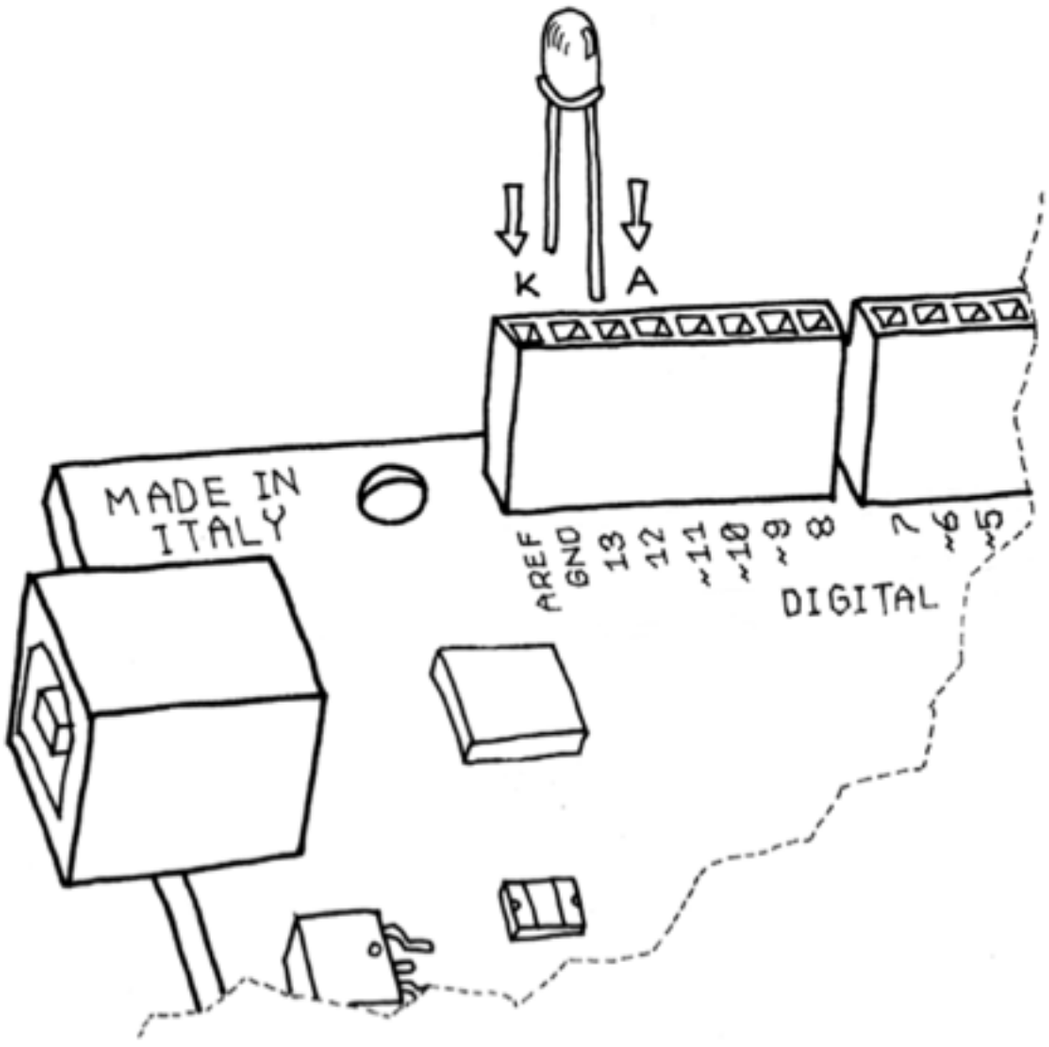
Arduino - Iniciação



- Config, Verify & Upload

LED - acender/apagar

~/ArduinoSynth_examples/Dia1_Efeitos/_01_LED_LigarDesligar.ino



retirado de "Getting Started with Arduino"

- Saídas digitais

LED - acender/apagar

~/ArduinoSynth_examples/Dia1_Efeitos/_01_LED_LigarDesligar.ino

Função setup()

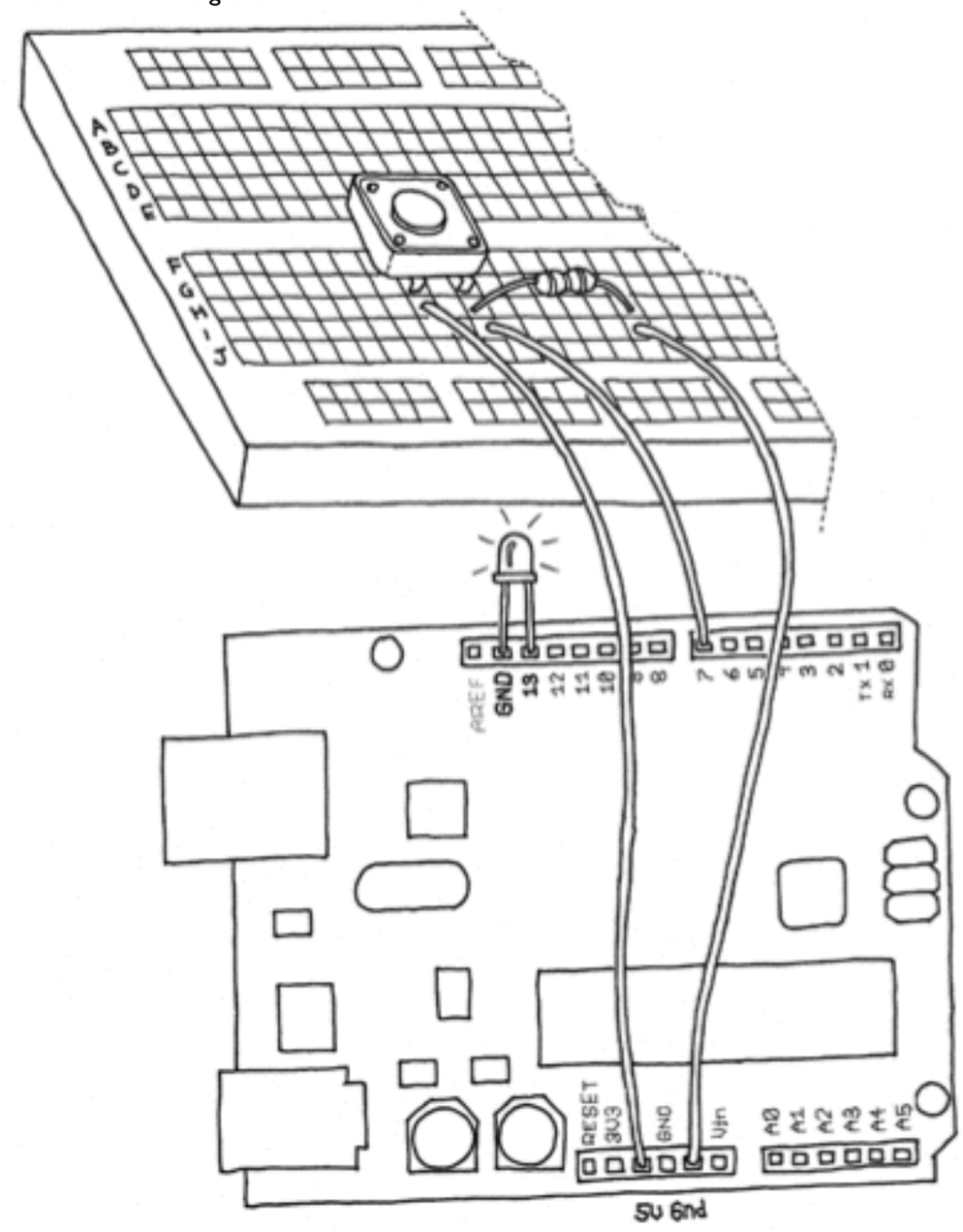
Função loop()

- Código Arduino (setup, loop, e outras funções)

Entradas Digitais

~/ArduinoSynth_examples/Dia1_Efeitos/_02_LED_ControloDigital.ino

retirado de "Getting Started with Arduino"



- Entradas digitais

Entradas Digitais

~/ArduinoSynth_examples/Dia1_Efeitos/_02_LED_ControloDigital.ino

_02_LED_ControloDigital | Arduino 1.6.1

_02_LED_ControloDigital

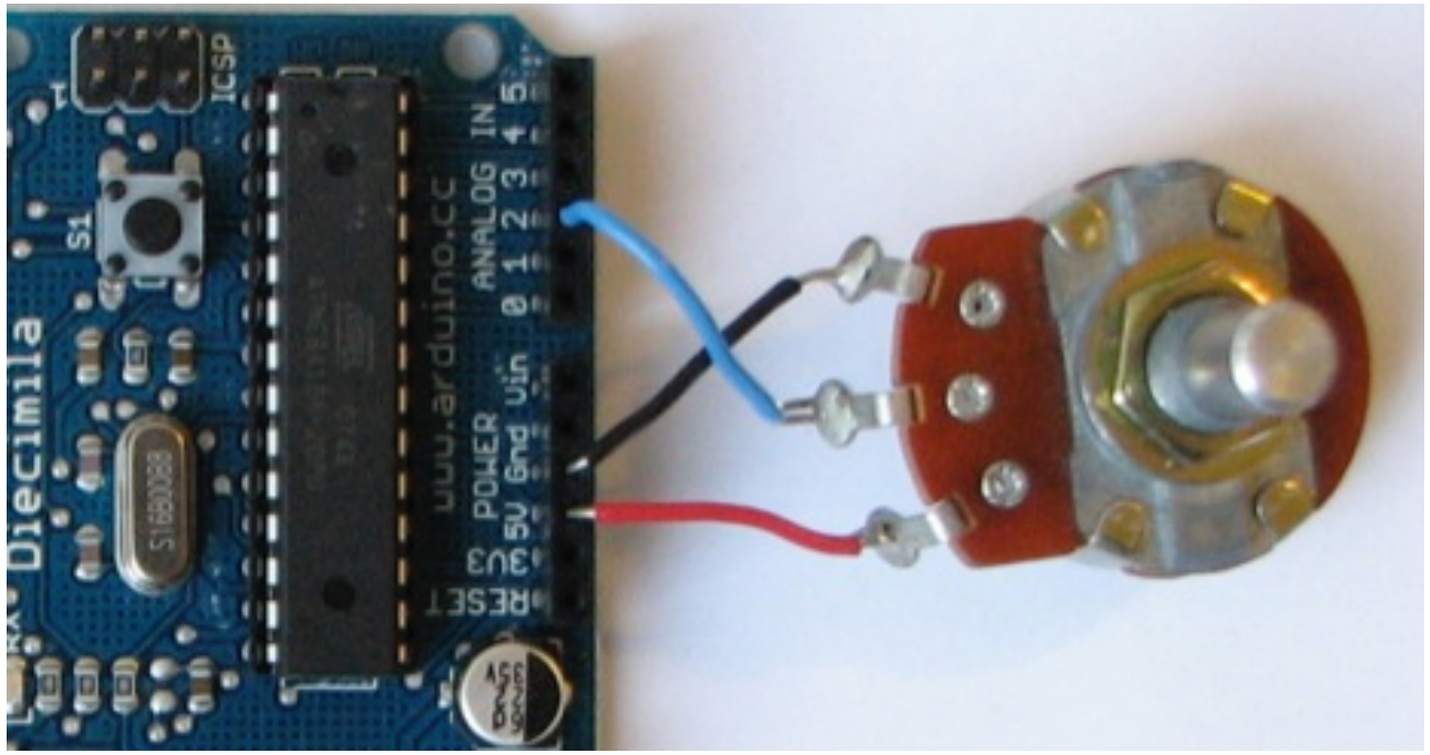
```
void setup() { //esta funçao e executada apenas uma vez quando o Arduino e reiniciado

  pinMode(13, OUTPUT); // define o pin 13 como um pin de saida

  pinMode(2, INPUT); // define o pin 13 como pin de entrada

}

// the loop function runs over and over again forever
void loop() { // esta funçao corre em loop repetidamente

  if (digitalRead(2) == HIGH){

    digitalWrite(13, HIGH);   // liga o LED

  }

  else {

    digitalWrite(13, LOW);    // desliga o LED

  }

}

// este exemplo e uma traducao de http://www.arduino.cc/en/Tutorial/Button
```

21 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on /dev/tty.usbmodem1411

- Lógica (funções *if* e *else*)

Entradas Analógicas

~/ArduinoSynth_examples/Dia1_Efeitos/_03_LED_ControloAnalogico.ino



retirado de http://www.arduino.cc/en/tutorial/potentiometer

- Entradas analógicas

Entradas Analógicas

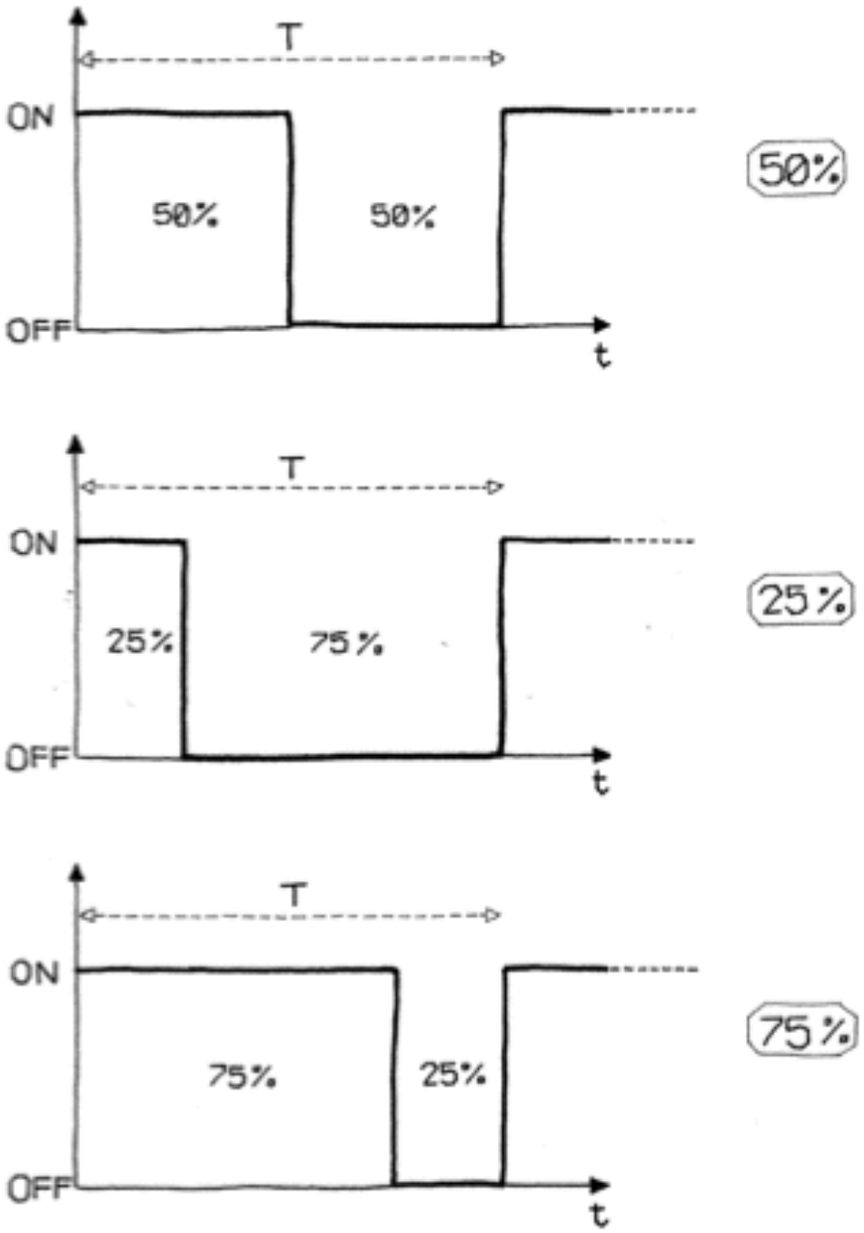
~/ArduinoSynth_examples/Dia1_Efeitos/_03_LED_ControloAnalogico.ino

```
/*
declara uma variavel do tipo "int" ( um numero inteiro )
e atribui-lhe o valor "0"
*/
int espera = 0;
```

- variáveis e declaração

Saídas digitais e PWM

~/ArduinoSynth_examples/Dia1_Efeitos/_04_LED_saidaPWM.ino



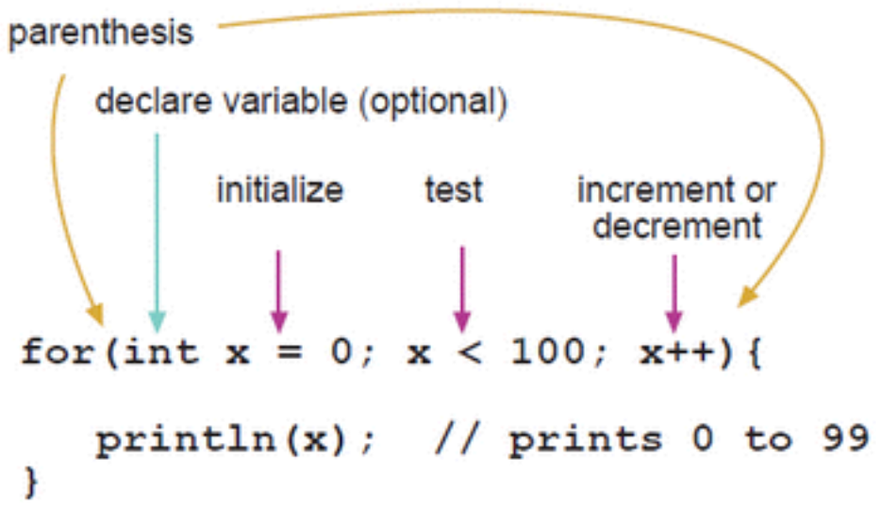
retirado de "Getting Started with Arduino"

- Pulse Width Modulation
- Saídas digitais PWM

Saídas digitais e PWM

~/ArduinoSynth_examples/Dia1_Efeitos/_04_LED_saidaPWM.ino

```
// rampa para acender o LED
for (i = 0; i < 255; i++){ // a funçao "for()" permite criar iteraçoes

  // "analogWrite()" permite-nos enviar valores intermedios entre "LOW" e "HIGH"
  analogWrite(LED, i);
  delay(10);

}

// rampa para apagar o LED
for (i = 255; i > 0; i --){
  analogWrite(LED, i);
  delay(10);
}
```



retirado de https://www.arduino.cc/en/Reference/For

- controlo e iteração

Leitura recomendada

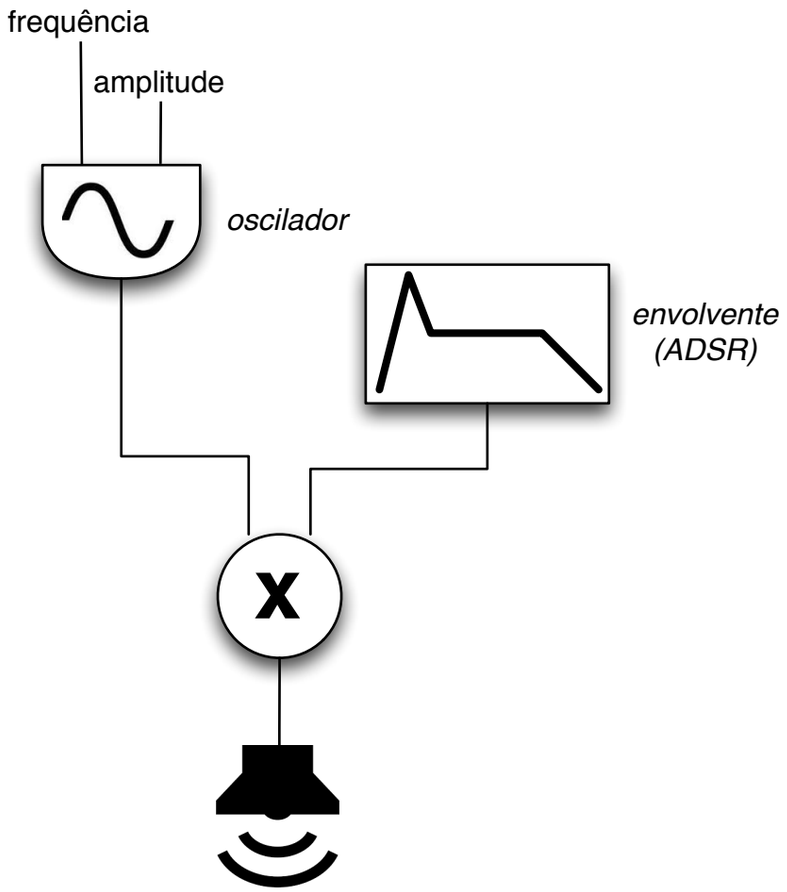
- Getting Started with Arduino (Massimo Banzi) : http://it-ebooks.info/book/1338/

- Getting Started in Electronics (Forrest Mims) : https://docs.google.com/file/d/0B5jcnBPSPWQyaTU1OW5NbVJQNW8/edit

- (Fórum Arduino : http://forum.arduino.cc/)

Mozzi

http://sensorium.github.io/Mozzi/

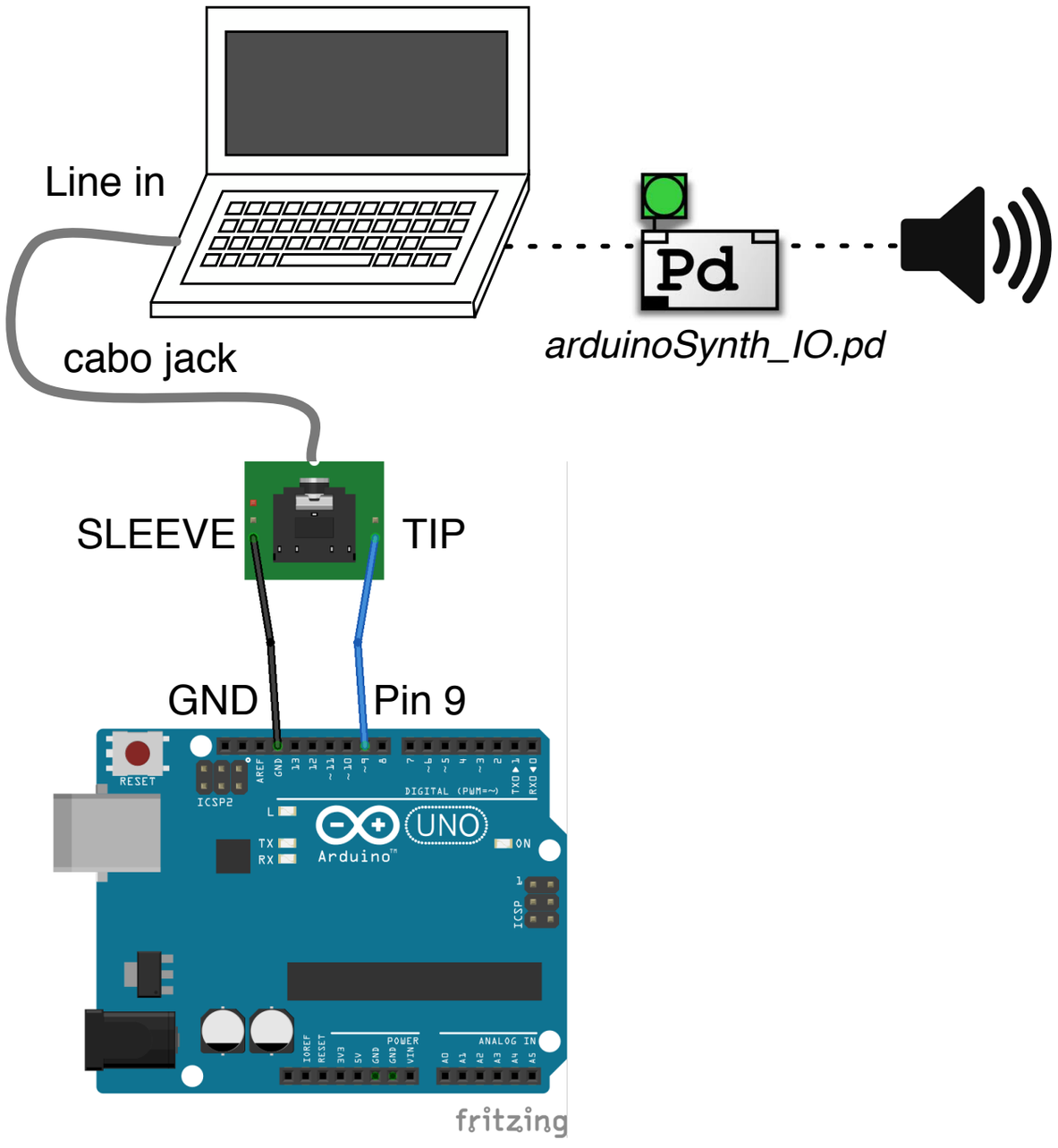


- Desenvolvida por Tim Barrass

- Biblioteca de síntese sonora para Arduino

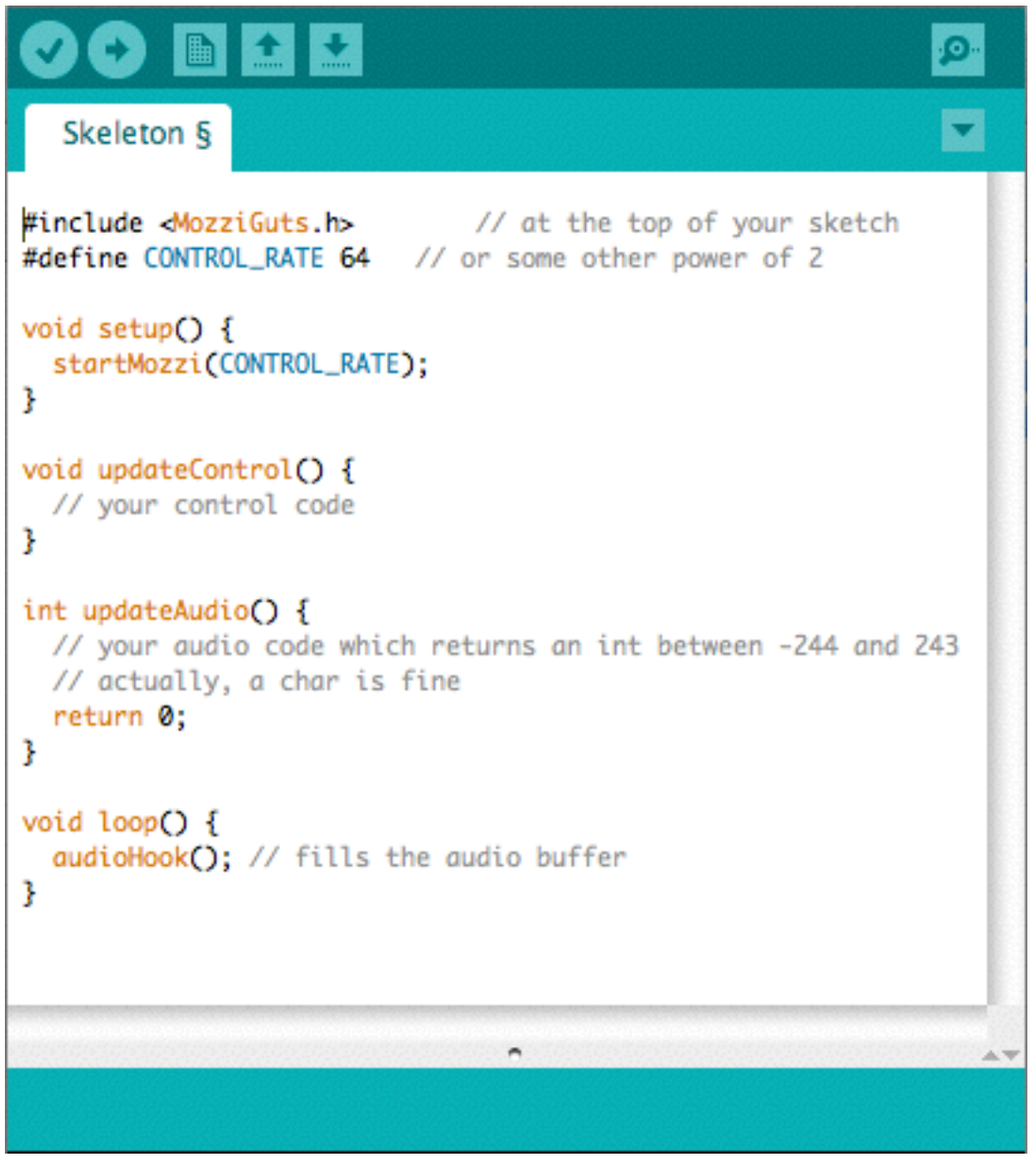
- O Arduino passa a ter osciladores, filtros, envolventes, delays, etc...

Iniciação e conexões



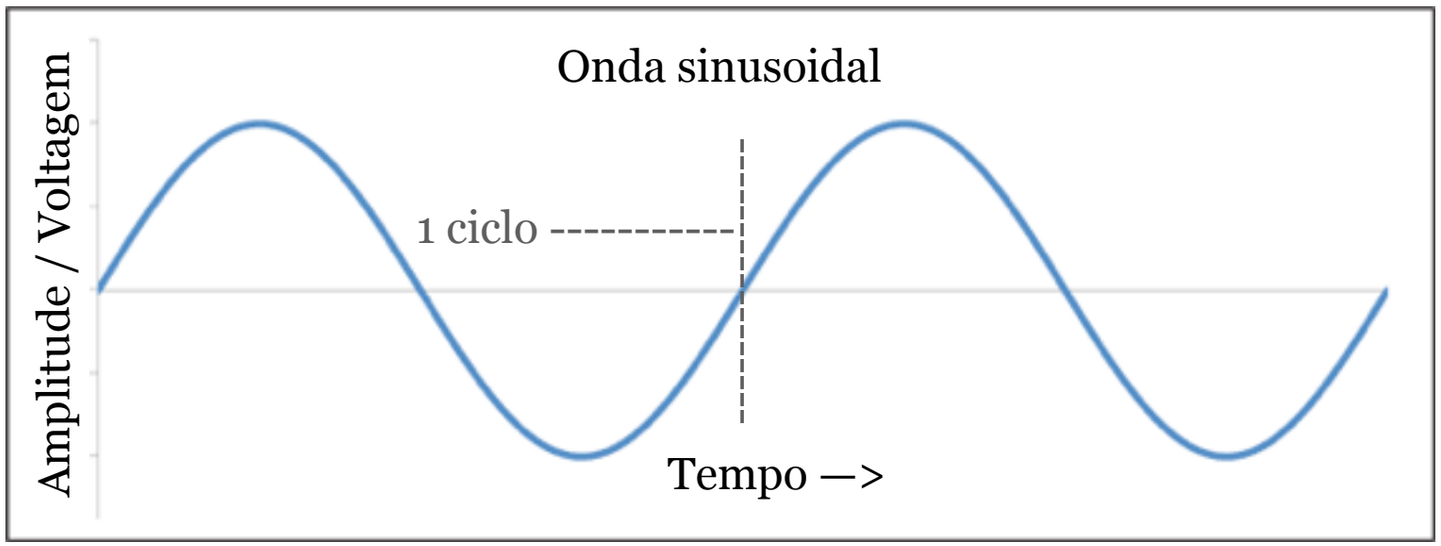
- Como instalar a biblioteca ?
- Carregar exemplo
- Como estabelecer uma ligação áudio ?

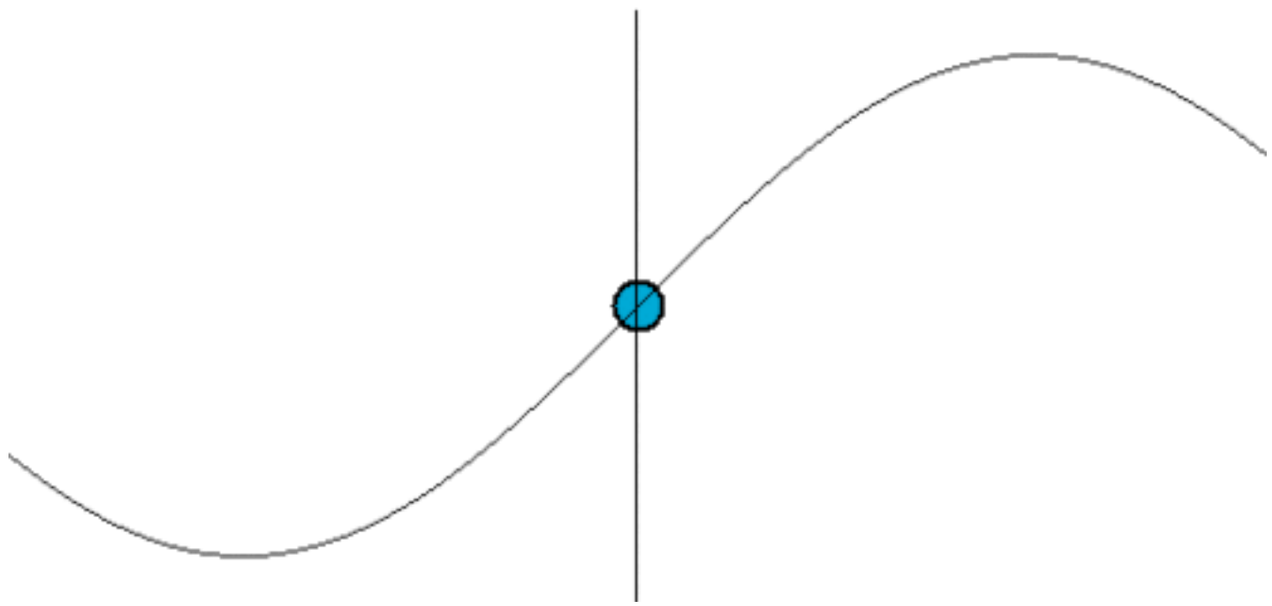
Arquitectura



- include Mozzi
- startMozzi + Control Rate
- updateControl
- updateAudio
- audioHook

Sinusoide

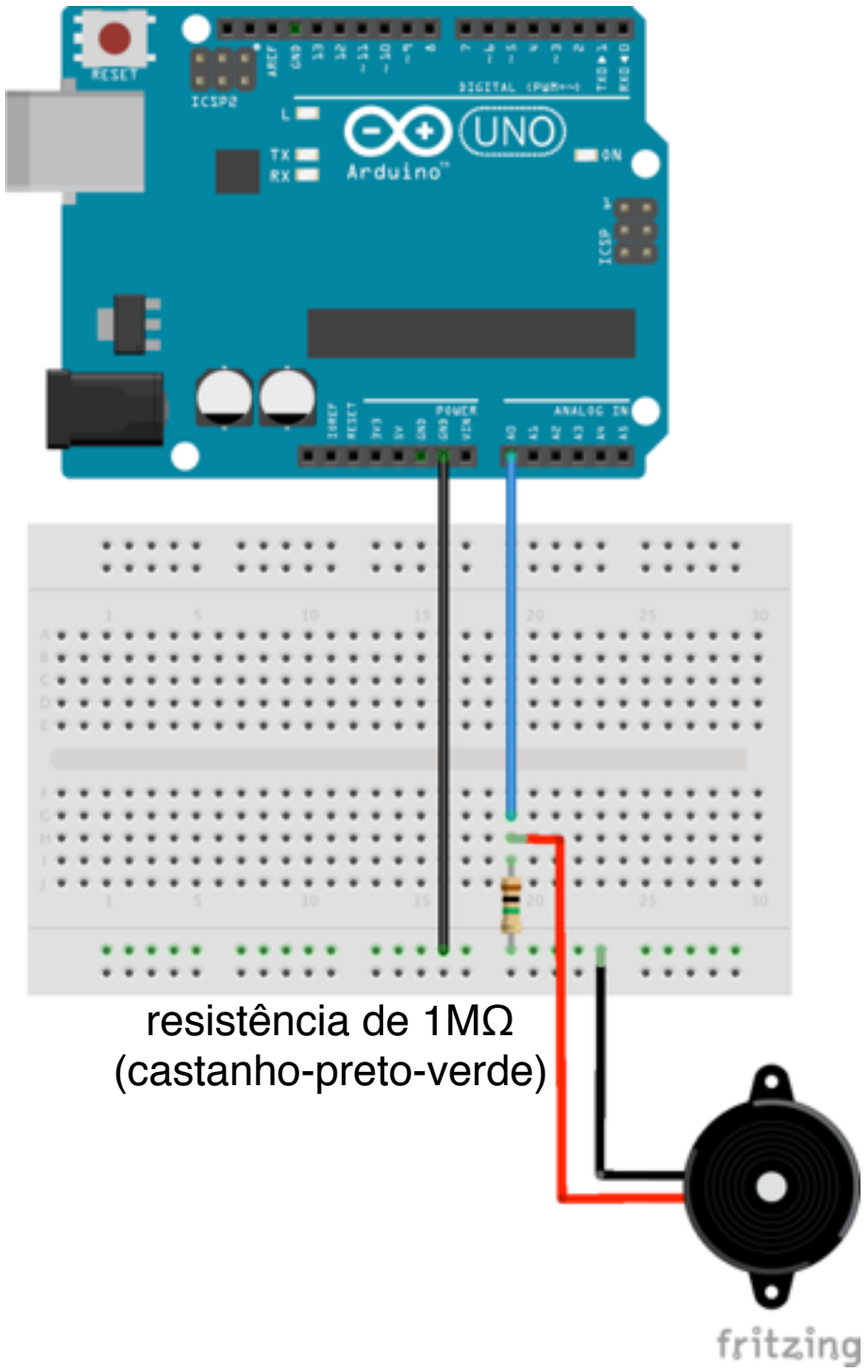




- O que são e como utilizar classes ?
- consultar doc. :

http://sensorium.github.io/Mozzi/doc/html/index.html

Piezo in



- conexões

- editar *mozzi_config.h*

```
72  #define USE_AUDIO_INPUT true
73  //#define USE_AUDIO_INPUT false
```

- função *getAudioInput()*

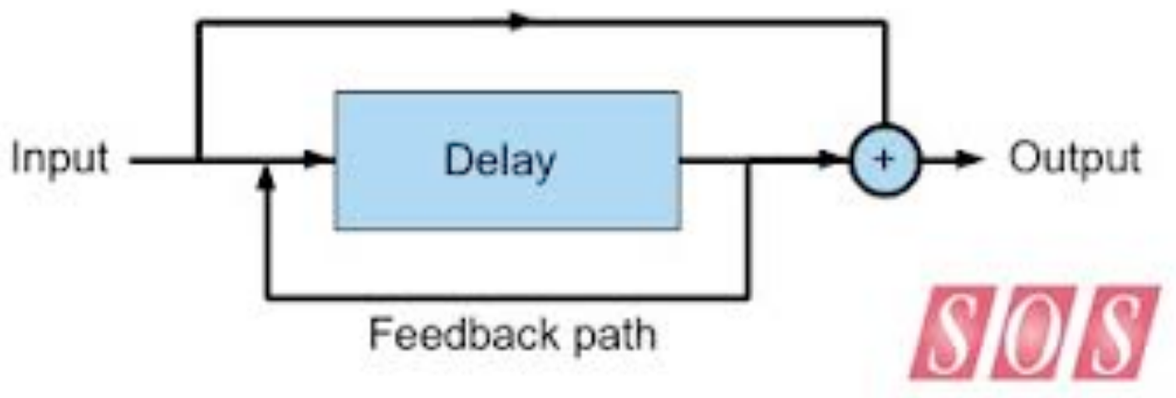
Optimização

```
26    //#define AUDIO_MODE STANDARD
27    #define AUDIO_MODE STANDARD_PLUS
28    //#define AUDIO_MODE HIFI
```

```
61    #define AUDIO_RATE 16384
62    //#define AUDIO_RATE 32768
```

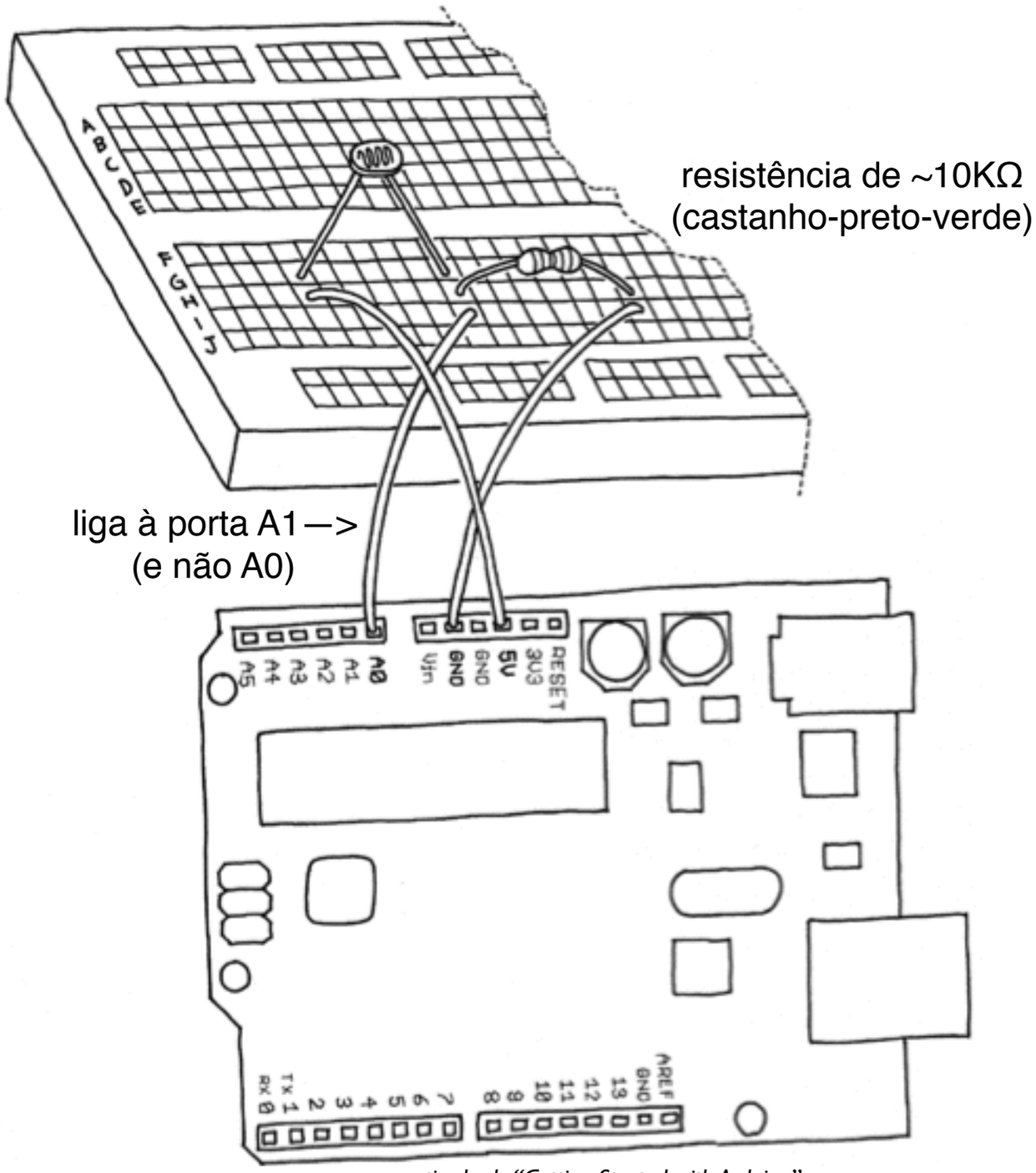
- *Output Modes (STANDARD, STANDARD_PLUS, HIFI)*
- *Output circuits (reference only)*

Feedback Delay



- AudioFeedbackDelay
- Parâmetros : duração do delay, ganho do feedback

Controlo



retirado de "Getting Started with Arduino"

- *Serial Monitor : Serial.begin()* e Serial.println()
- *Lowpass filter*
- Parâmetros : frequência de corte, ressonância/ resposta
- Outras funções : *IntMap*

Outros Efeitos

- Waveshaping // distorção
- Flanger (feedback delay + modulação no tempo de duração)
- bit distortion (usando bitshif >> e <<)
- AM // tremolo
- Reverb
- .....

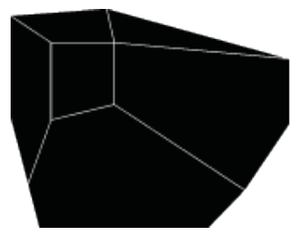
Extras

- Output modes (standard, standard plus e HiFi) e circuitos de saída
- Entrada áudio adequada (http://www.instructables.com/id/Arduino-Audio-Input/?ALLSTEPS)
- Modulações (osciladores, random, etc...)
- Hints & Tips (variáveis, funções)
- Algoritmos: www.musicdsp.org

Arduino Synth

Sintetizadores DIY

tiago.a.s.angelo@gmail.com



casa da música

DIGITÓPIA

Plano (domingo)

Manhã

Overview - modelos de síntese

Experimentação

Tarde

Controlo MIDI

Hands On - construção de um sintetizador

Síntese

- modelos de síntese
  - abstractos/modulação
  - gravações processadas
  - *espectrais*
  - *físicos*
  - ref.: *Theory and Technique of*
    e
- contexto histórico / exemplos

REF.: *Theory and Technique of Electronic Music (Miller Puckette)*

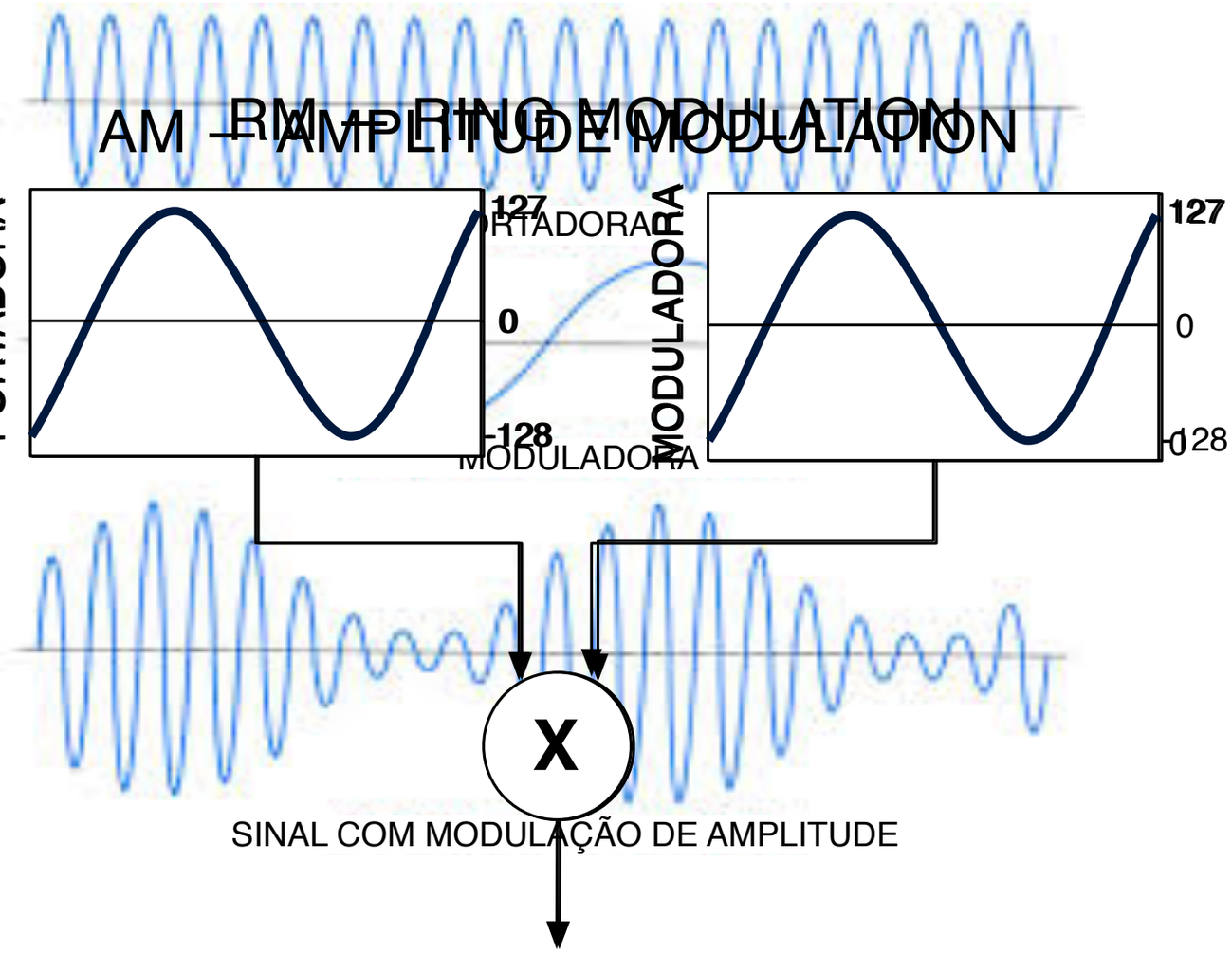
Tabelas e osciladores

- Síntese *wavetable*
- Formas de onda: Documentos/Arduino/libraries/Mozzi/tables/
- As tabelas têm uma resolução de amplitude de 8-bits, variando entre -128 e 127

```
/*
  Basicamente um oscilador le uma tabela
  de valores que representam uma forma de onda
*/

// Incluimos a classe 'Oscil' para usar osciladores
#include <Oscil.h>
// Incluimos a tabela que queremos usar
#include <tables/sin2048_int8.h>

//Declaramos o oscilador
Oscil <SIN2048_NUM_CELLS, AUDIO_RATE> oMeuOscilador(SIN2048_DATA);
//< Tamanho da tabela, Quantas x faz update> nomeDoOscilador(Dados para o oscilador)
```

_01_SYNTH_ModulacaoAmplitude.ino

Modulação de amplitude (AM/RM)



- Usa um sinal modulador para alterar a amplitude de um sinal portador
- Tremolo (freq. moduladora < 16Hz)
- AM - moduladora unipolar
- RM - moduladora bipolar

DICA:

Modulation Depth = amplitude da moduladora

_02_SYNTH_AdditiveSynth.ino

Síntese Aditiva

- segundo Fourier é possível uma onda complexa pode ser reconstruida através da adição de várias ondas sinusoidas

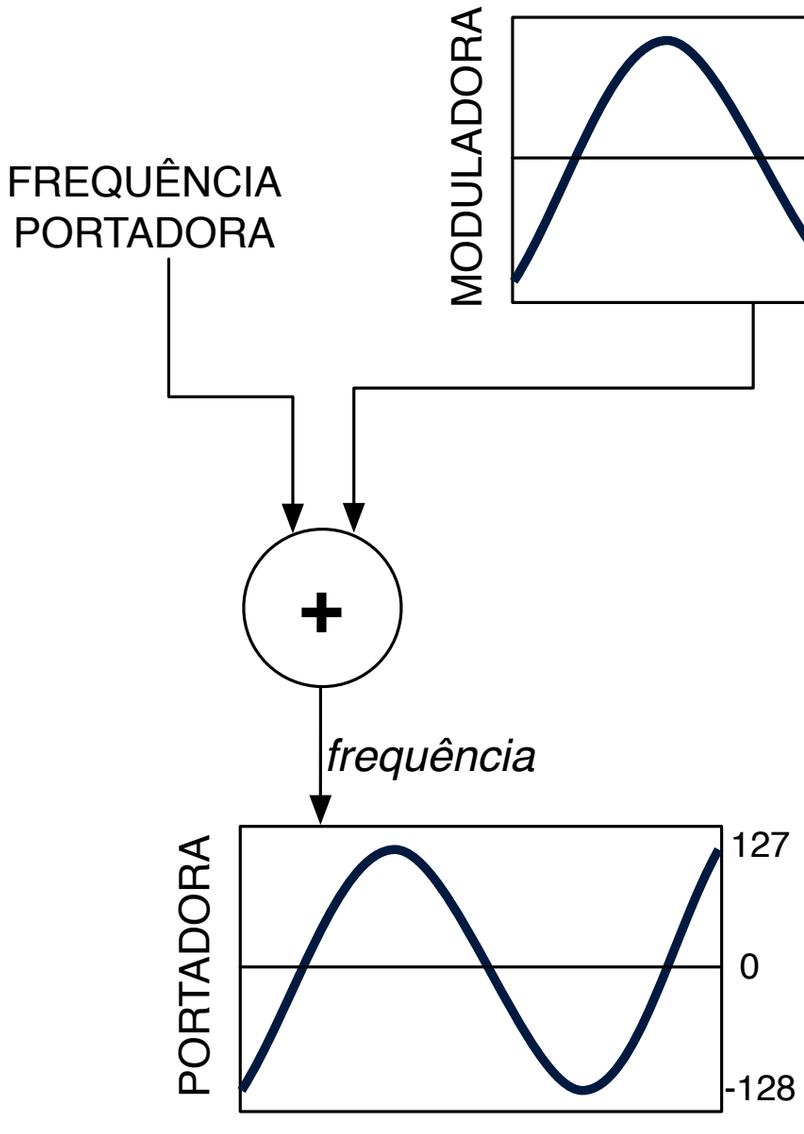
- este tipo de síntese consiste basicamente na adição de vários osciladores

- o timbre é então determinado pela amplitude e frequência de cada oscilador, sendo que frequências próximas poderão criar batimentos, ou seja, as frequências interagem entre si criando depressões e aumentos de amplitude

- ver também exemplo Mozzi "Detuned_Beats_Wash.ino"

_03_SYNTH_ModulacaoFrequencia.ino

Modulação de Frequência (FM)



- Descoberta por John Chowning em 1967
- O seu ponto forte era a possibilidade de criar tons complexos (semelhante à sintese aditiva) com apenas dois osciladores

_04_SYNTH_SimpleSampler.ino

Sampling

- Os samples são em tudo semelhantes às tabelas, mas por norma são mais longos (>NUM_CELLS)

- Estão em Documentos/Arduino/libraries/Mozzi/samples/

- Tal como os osciladores também é possível alterar a frequência!

MIDI

- o que é ?
- tipos de mensagens (pitch, vel, cc, pg ch, pitch bend...)
- formato das mensagens

MIDI no Arduino (usb)

1. Carregar o sketch _05_MIDI_SimpleSineNoteOn.ino
2. Abrir o "Hairless midiserial"
3. Nas preferências do Hairless midiserial certifique-se que o baud rate está a 9600
4. Depois escolha a porta MIDI In de onde vai enviar MIDI
5. Escolha a Serial port relativa ao Arduino (aqui os nomes são diferentes do que aparece na porta do Arduino IDE)

ATENÇÃO: não é possível carregar código para o Arduino enquanto o Hairless midiserial estiver ligado pois a porta de comunicação fica ocupada

Control Change (cc)

- como receber MIDI (e enviar)
- *como criar uma entrada MIDI ?*
- enviar MIDI por USB

Mozzi, MIDI e Funções desactivadas

- Funções (mtof, ftom...)
- ATENÇÃO!!! O Mozzi desactiva as seguintes funções do Arduino:
  - delay(), delayMicroseconds(), millis() e micros()
  - em substituição tem: EventDelay(), Metronome() e mozziMircros()

+ Síntese

- additive synthesis (Detuned_Beats_Wash.ino)
- PWM phasing (PWM_Phasing.ino)
- Phase distortion (PDresonant.ino)
- scrubbing
- granular ?

REF.: “Computer Music Tutorial”,

Extras

- sequências
- standalones (alimentação,...)
- Optimização do código (FixedMath e variáveis Mozzi...)
- Debug
- como criar tabelas (soundtables) ?
- *cv control in/out ?*
- como ler diagramas ?
- circuitos electrónicos pos/pre Arduino synth (filtros, spring reverb, ?)