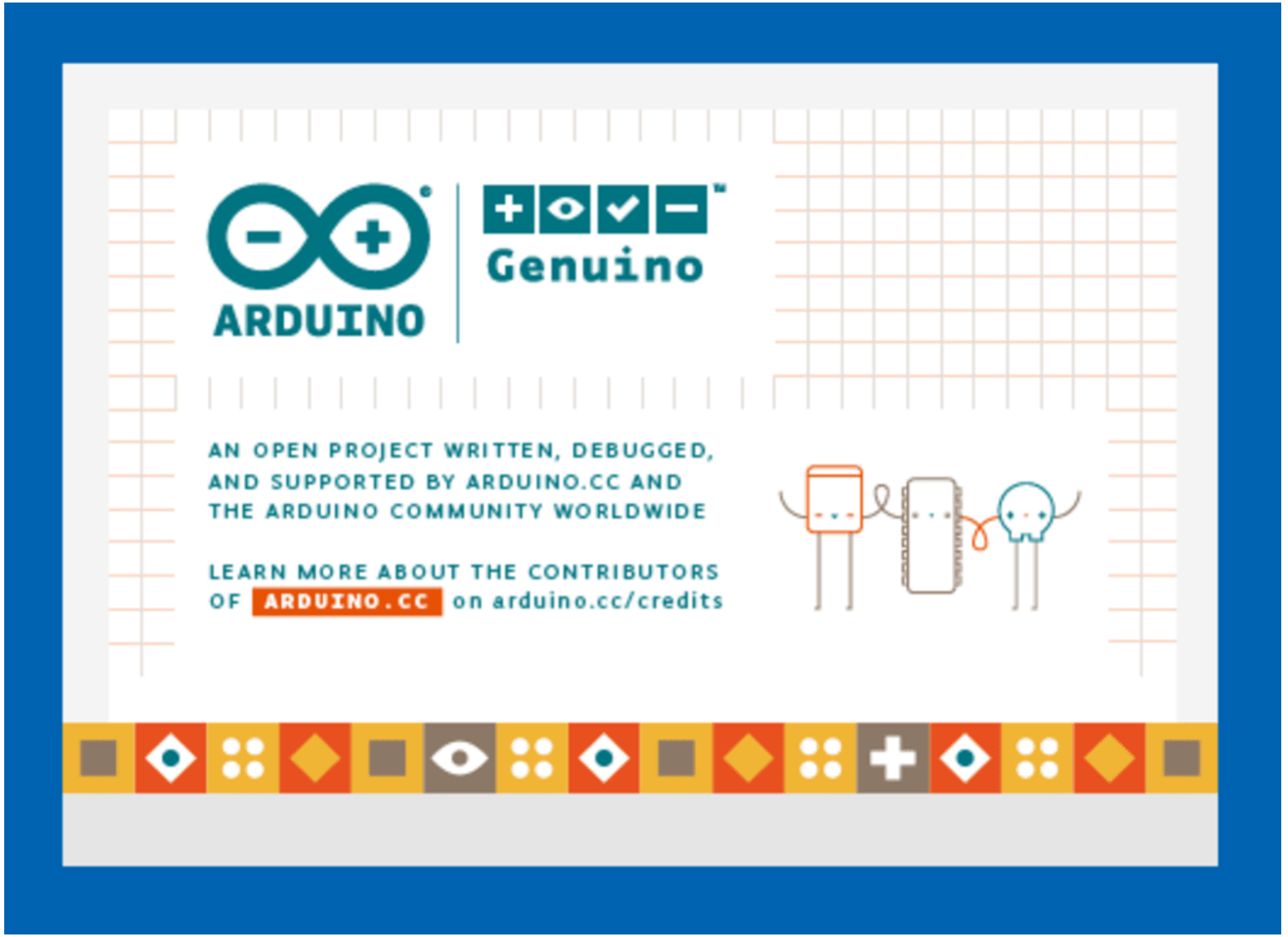
Arduino

Interação Pessoa Máquina



Conceito

- Plataforma de prototipagem eletrónica open source
- Hardware e software flexíveis, modular, extensíveis e fáceis de usar
- Ambiente de desenvolvimento multiplataforma
- Destinado a todos os interessados na criação de protótipos, invenções e todo o tipo de artefactos
  - Concretizar ideias e conceitos
  - Incentivar a prototipagem
  - Incentivar aprendizagem autónoma
  - Incentivar à experimentação

Origem

- Surge em 2003, em Itália
- Resultado de uma Tese de Mestrado de Hernando Barragán, sob a orientação de Massimo Banzi e Casey Reas.
- O projeto Arduino foi iniciado por Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis.
- O objetivo foi o de elaborar uma Plataforma que fosse ao mesmo tempo barata, funcional e fácil de programar, sendo dessa forma acessível a estudantes e amadores.

Uma família de dispositivos...

Arduino Uno

Arduino Leonardo

Arduino Nano

Arduino Duemilanove

Arduino Due

Arduino Mega ADK

Arduino Ethernet

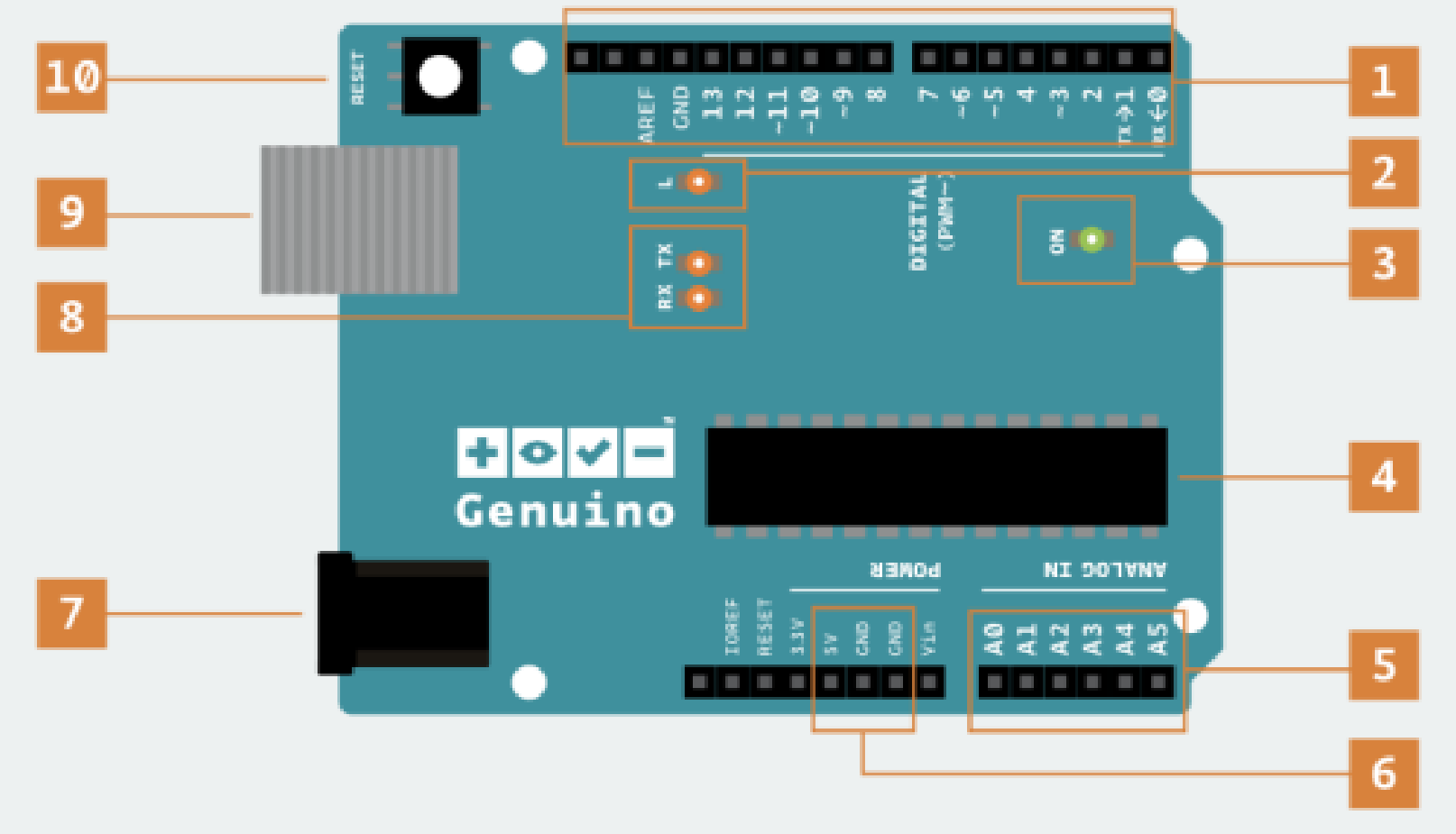
Arduino Yun

Arduino Galileo

Microcontrolador

- Computador de dimensões reduzidas num circuito eletrónico integrado contendo uma unidade de processamento, memória e periféricos programáveis de entrada/saída.
- Usado em dispositivos embebidos para desempenhar tarefas muito específicas

Anatomia


10
9
8
7
1
2
3
4
5
6
RESET
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
L
TX
RX
DIGITAL (PWM~)
ON
Genuino
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

Anatomia

- 1. Pinos Digitais, usam-se com funções digitalRead(), digitalWrite()
- 2. Pino 13 LED
- 3. Pino de energia (power)
- 4. Microcontrolador ATmega (unidade processamento)
- 5. Pinos Analógicos, usando as funções analogRead(), analogWrite()
- 6. Pinos GND e 5V, usam-se para alimentação
- 7. Ligação de energia
- 8. LEDs RX e TX
- 9. Entrada USB
- 10. Botão de reset do microcontrolador

Instalação

- Ir a https://www.arduino.cc/en/Main/Software
- Fazer download da última versão de acordo com o sistema operativo
- Proceder com à instalação

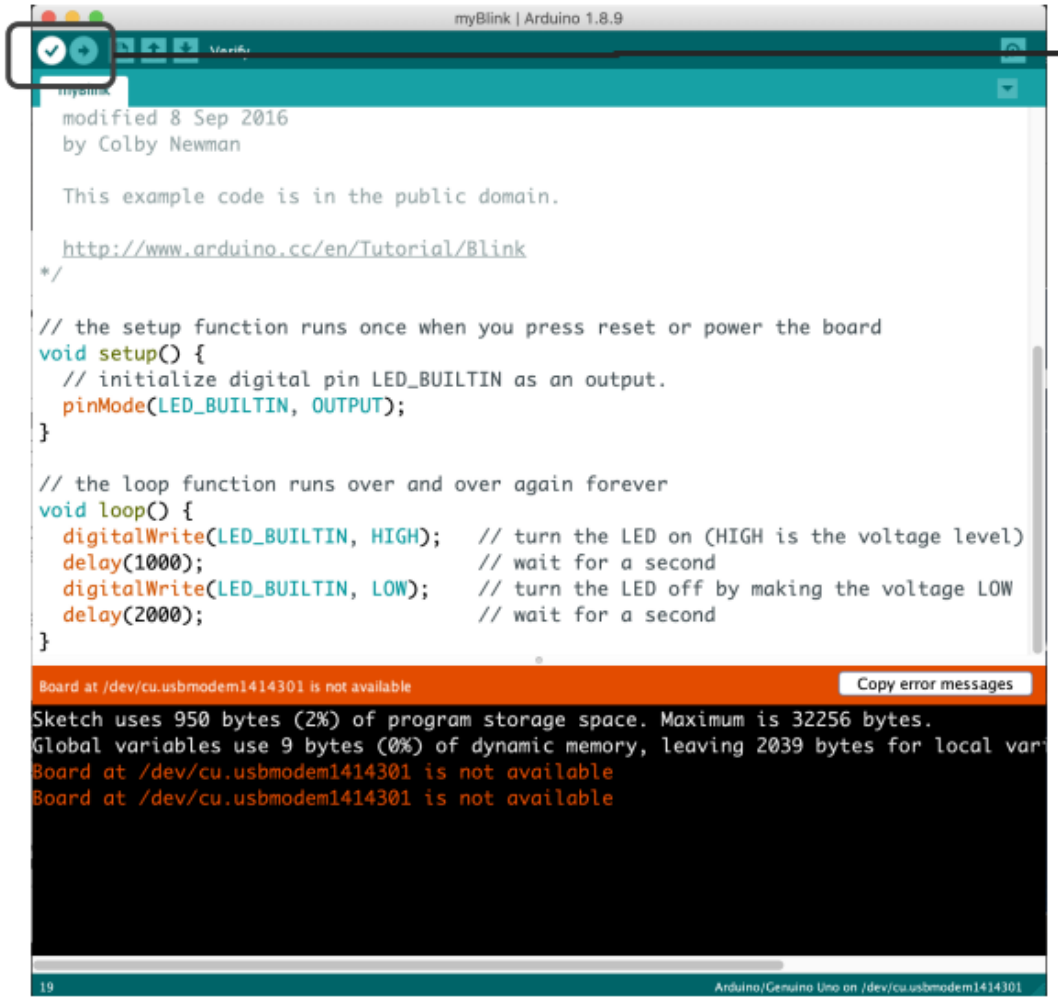
- http://arduino.cc/en/Guide/HomePage

Ligação da placa com o IDE

- Ligar o cabo USB ao computador e à placa Arduino
- Executar a aplicação Arduino IDE
- Menu Tools -> Board (Escolhar o modelo Arduino)
- Menu Tools -> Port (Escolher a porta de comunicação ond eo Arduino está ligado)

- Ver exemplos (File->Examples)

IDE



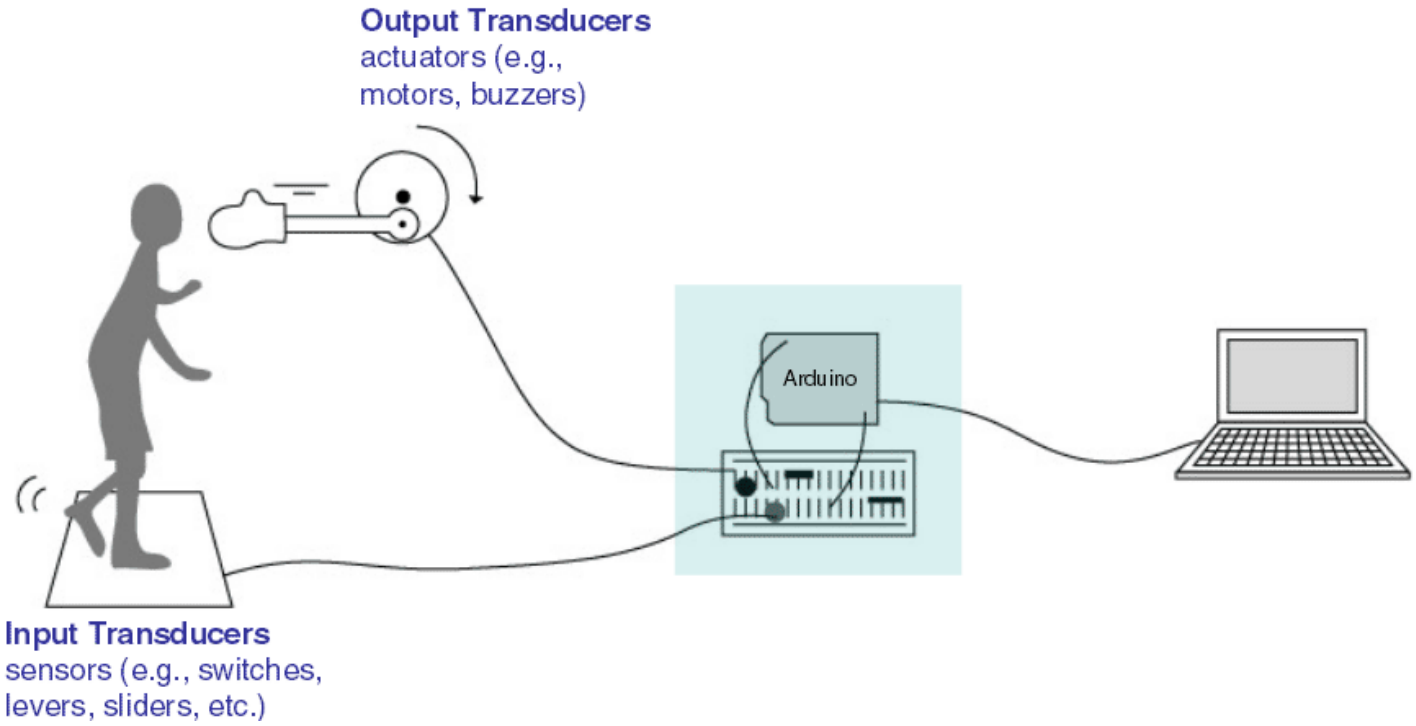
Botões de compilação e execução do programa na placa Arduino

Janela de edição do código

Consola com informação do estado, erros, avisos das operações

Input/Output

- Image from *Theory and Practice of Tangible User Interfaces* at UC Berkley



Como funciona?

- A lógica de um programa para Arduino encontra-se sempre organizada em 2 blocos fundamentais:
  - O bloco **setup**, que apenas é executado uma vez, sendo utilizado para iniciar os modos do hardware, variáveis ou qualquer outra coisa que necessite de definição de um estado inicial
  - É chamado apenas quando o Arduino é ligado ou reiniciado. É usado para iniciar variáveis e modos de pinos

```
void setup() {
    // put your setup code here, to run once:
}
```

  - O bloco **loop**, que é executado indefinidamente até que o Arduino seja desligado da alimentação

```
void loop() {
    // put your main code here, to run repeatedly:
}
```

Programação

- Funções
  - digitalWrite() - Arduino Reference
  - digitalRead() - Arduino Reference
  - pinMode()
- Estruturas
- Variáveis
- Arduino Reference - Arduino Reference

PinMode

- Um pino no arduino pode ser colocado como input ou output usando a função pinMode.

- pinMode(13, OUTPUT); // sets pin 13 as output pin

- pinMode(13, INPUT); // sets pin 13 as input pin

Reading/writing digital and analog values

- digitalWrite(13, LOW); // Makes the output voltage on pin 13 , 0V
- digitalWrite(13, HIGH); // Makes the output voltage on pin 13 , 5V

- int buttonState = digitalRead(2); // reads the value of pin 2 in buttonState

- analogRead(A0); // used to read the analog value from the pin A0
- analogWrite(2,128);

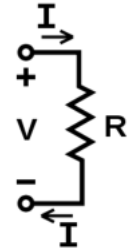
Analog to Digital Coversion

- Analógico
  - It is continuous range of voltage values (not just 0 or 5V)
- Digital
  - Because our microcontroller only understands digital.
- The Arduino Uno board contains 6 pins for ADC
- 10-bit analog to digital converter
- This means that it will map input voltages between 0 and 5 volts into integer values between 0 and 1023

Como funciona?

- De acordo com a sua *data sheet*, um LED necessita de 1,8 a 3V e tem uma corrente direta média de 20 a 25 mA.
- Como estamos a utilizar os 5V do Arduino, devemos colocar uma resistência em série com o LED.
- De acordo com a Lei de Ohm, precisamos de uma resistência de 220 Ohm (código: vermelho, vermelho, castanho).

V=R.I



Como funciona?

- Um LED vermelho típico tem uma queda de tensão de 1.8V e uma corrente direta (forward current) de cerca de 20-25mA.
- O pino do Arduino produz uma tensão de saída de 5V. V = (voltagem do arduino) – (queda de tensão do LED) = 5V – 1.8V = 3.2 V
- I (corrente direta do LED) = 20 mA R=V/I = 160 Ohm. Mas não temos resistências de 160 Ohm !
- Valores comercialmente disponíveis: {100, 220, 470, 1000, 2200, 4700, 10000, … }. Escolhemos o valor disponível imediatamente acima dos 160 Ohm.

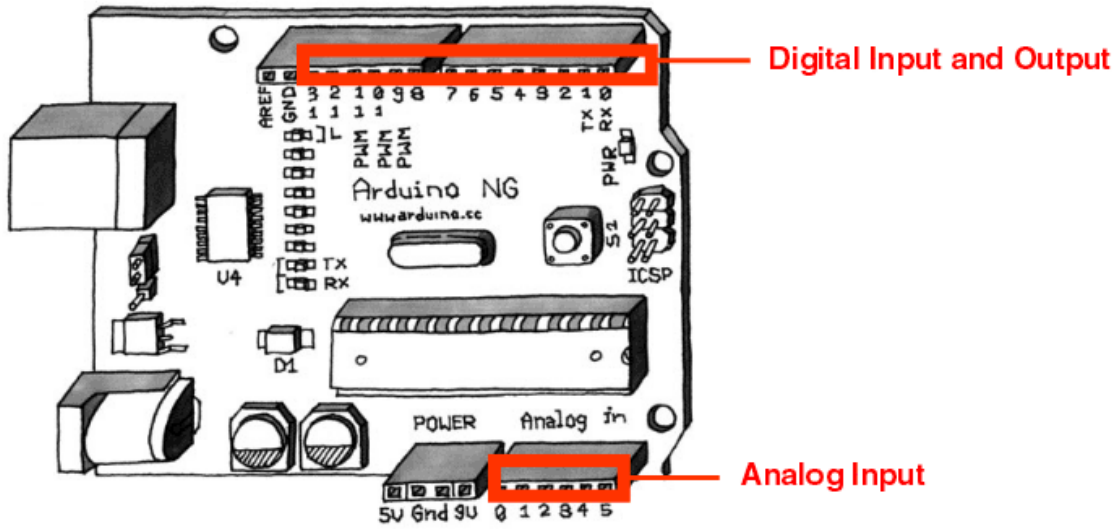
Como funciona?

- Quando se subsitui R=220 na equação I=V/R I = V/R, I = 3.2/220 ~= 14mA, o que ainda é aceitável. Iremos obter um valor da corrente em torno de 14mA.
- Os LEDs operam entre 10- 25mA. Além disso, como os LEDs são dispositivos não-lineares, a diferença entre uma corrente de 14mA e 25mA não significa necessariamente uma diferença proporcional no brilho. Na maioria dos casos, é provável que não se note diferença… Vamos então utilizar uma resistência de 220 Ohm (código: vermelho, vermelho, castanho).

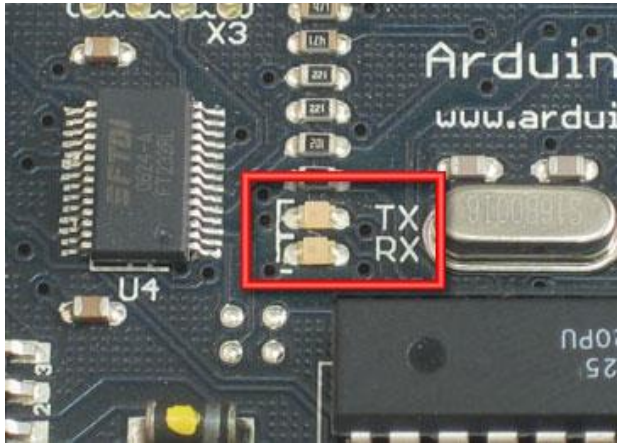
IO Pins

- Image from *Theory and Practice of Tangible User Interfaces* at UC Berkley

Two states (binary signal) vs. multiple states (continuous signal)



Serial Communication



- ***Compiling*** turns your program into binary data (ones and zeros)
- ***Uploading*** sends the bits through USB cable to the Arduino
- The two LEDs near the USB connector blink when data is transmitted
  - **RX** blinks when the Arduino is receiving data
  - **TX** blinks when the Arduino is transmitting data

Some Commands

Example Program

- Serial.begin()
  - e.g., Serial.begin(9600)
- Serial.print() or Serial.println()
  - e.g., Serial.print(value)
- Serial.read()
- Serial.available()
- Serial.write()
- Serial.parseInt()

Tinkercad

- https://www.tinkercad.com/things/
- O Tinkercad é um simulador online gratuito da Autodesk, onde podem experimentar praticamente tudo com o Arduino, de modo seguro.
- Pode ser também utilizado para testar circuitos e código antes de implementar.
- É vos-recomendado que se registem e testem os exemplos, explorando on interface do Tinkercad.

Exemplos de projetos

- https://www.instructables.com/Arduino-PowerPoint-Pointer/
- Kit Maker Arduino: Introdução - FilipeFlop

- Projeto 1 - Pisca Pisca
- Projeto 2 - SOS Luminoso
- Projeto 3 - Brilho oscilante
- Projeto 4 - Semáforo
- Projeto 5 - Interruptor de luz
- Projeto 6 - Luzes Coloridas
- Projeto 7 - Troque a cor das luzes
- Projeto 8 - Acionando um Motor
- Projeto 9 - Controlando o motor
- Projeto 10 - Sensor de luz ambiente
- Projeto 11 - Dó Ré Mi
- Projeto 12 - Alarme com Sensor a Laser
- Projeto 13 - Alarme de movimento
- Projeto 14 - Contador Digital
- Projeto 15 - Dado Eletrônico