Arduino Básico Projetos em Arduino usando simuladores

Perfil

Nilson Ramos de Menezes Júnior
nilson.junior@dcomp.ufs.br

nilson.ramos@souunit.com.br

www.linkedin.com/in/nilojr90

CÓDIGO DE TREINADOR :
6385 0247 1651

Nilson Ramos é graduando em Sistemas de Informação, pela Universidade Federal de Sergipe onde atua como Técnico em Informática desde 2013. Eletricista predial, pelo SENAI Sergipe. E por curiosidade desmonta os aparelhos eletrônicos da família desde a década de 90. Seu interesse pela plataforma Arduino surgiu dessa curiosidade.

erbase 2018

http://bit.ly/arduino18erbase

Objetivos

Apresentar o Arduino, como uma plataforma viável para a prototipação e desenvolvimento de projetos.

Mostrar as funções básicas da linguagem de programação do Arduino.

Apresentar conceitos básicos de eletrônica analógica e digital.

Pré Requisitos

Conhecimento prévio de alguma linguagem de programação, ou programação por blocos.

Uma conta em qualquer um desses serviços:

Facebook, Google, Yahoo, Microsoft ou Autodesk.

Introdução

Você pode dizer à sua placa o que fazer enviando um conjunto de instruções para o microcontrolador na placa.

Para fazer isso, você usa a linguagem de programação Arduino.
[https://www.arduino.cc/en/Guide/Introduction]

Para esse minicurso usaremos como referência o Arduino Uno.

https://www.arduino.cc/en/Main/Products

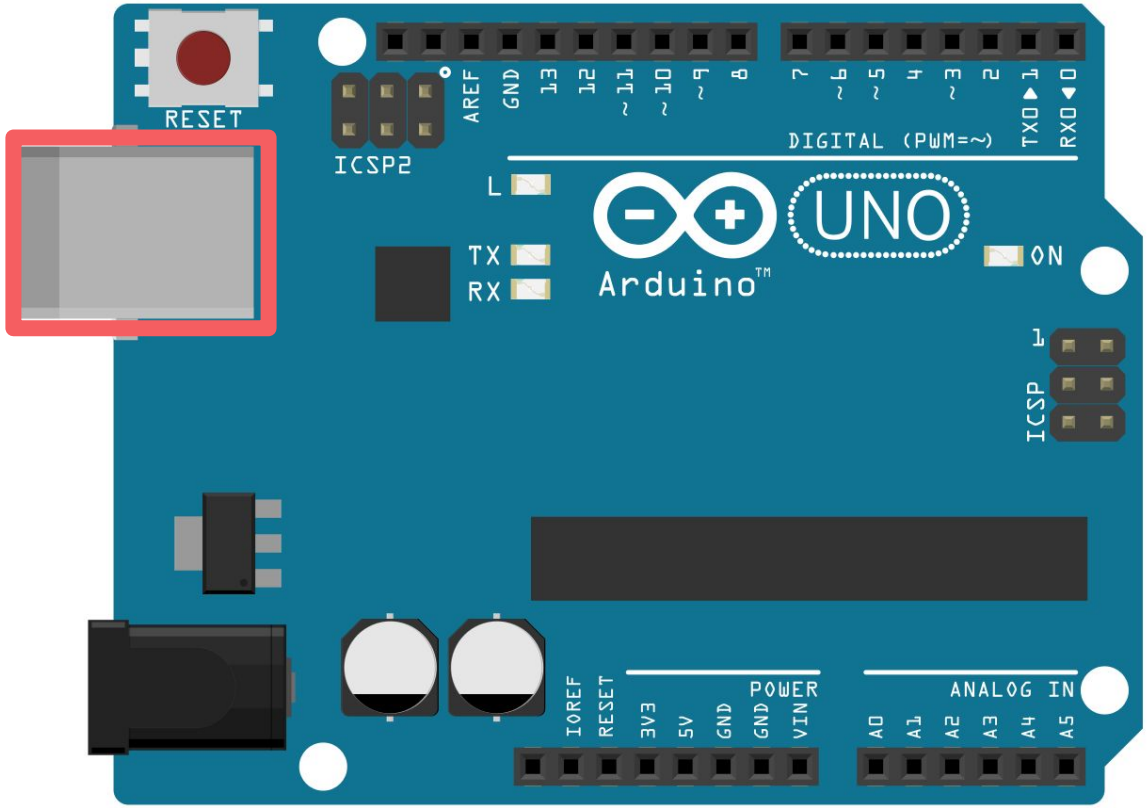
Anatomia do Arduino Uno

https://www.arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf

- Visão geral
- O Microcontrolador ATmega.
- Referência dos pinos (pinout).

Conexão USB

Alimentação
Porta serial para gravação dos sketch.*

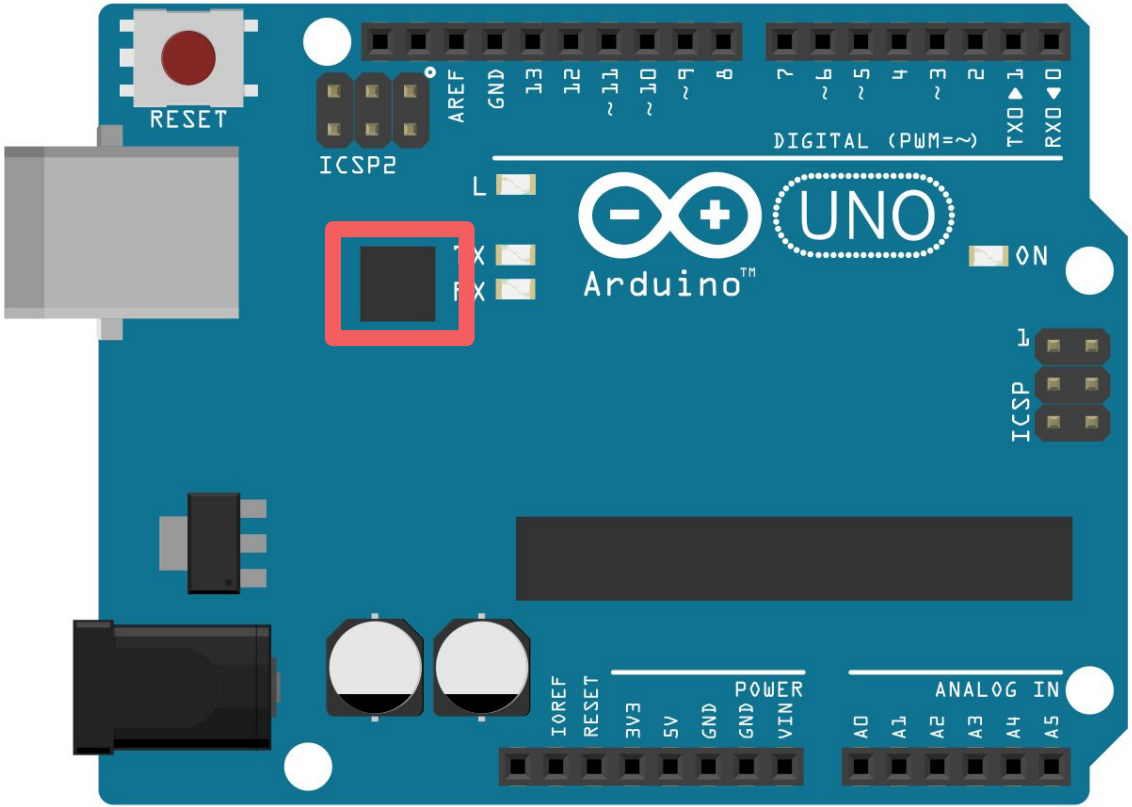


Controlador USB-Serial

https://www.arduino.cc/en/Guide/DriverInstallation

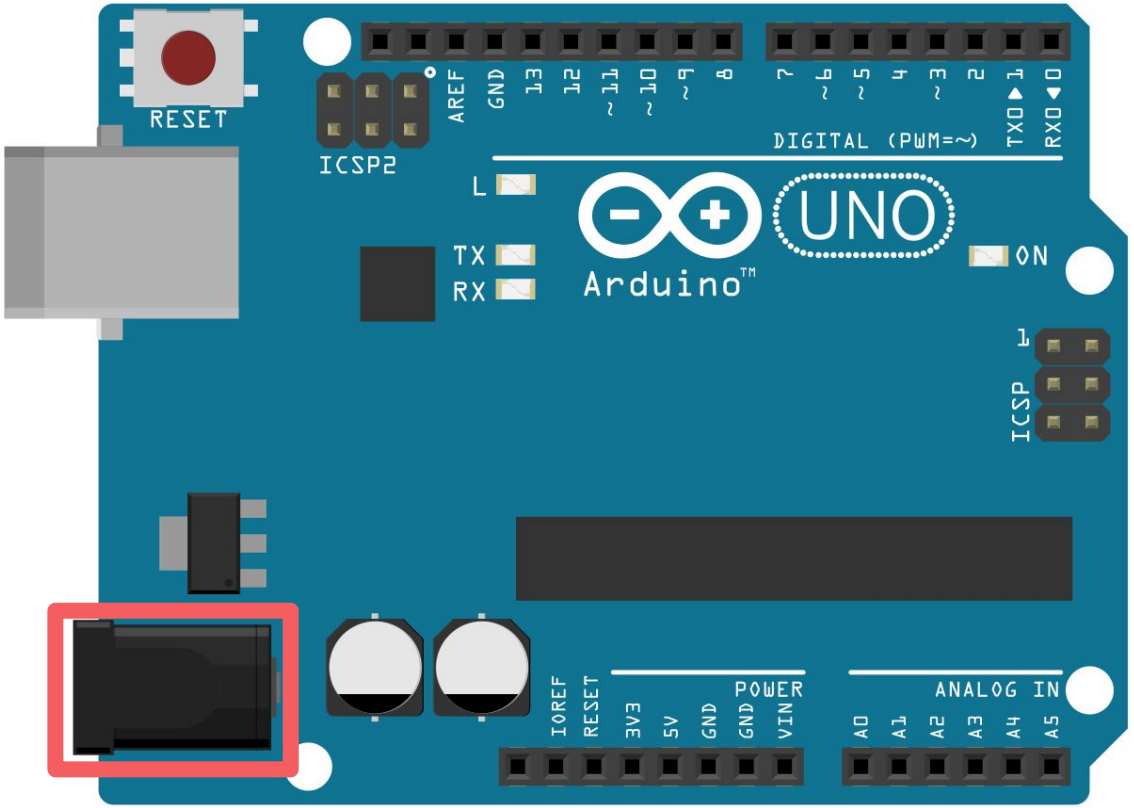
CH340 Driver : http://www.arduined.eu/files/CH341SER.zip

ATmega16u2



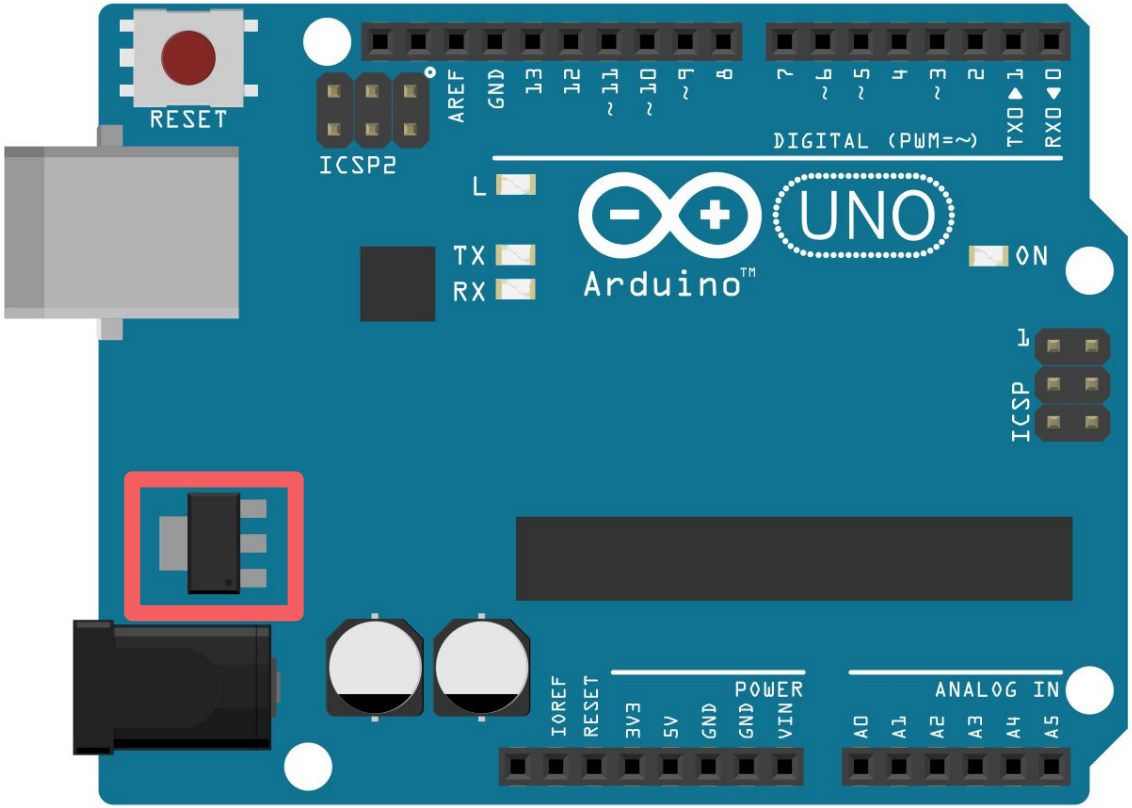
Conector de Alimentação

6v a 20v
recomendado **7v a 12v**



Regulador de Tensão

http://www.onsemi.com/PowerSolutions/product.do?id=NCP1117



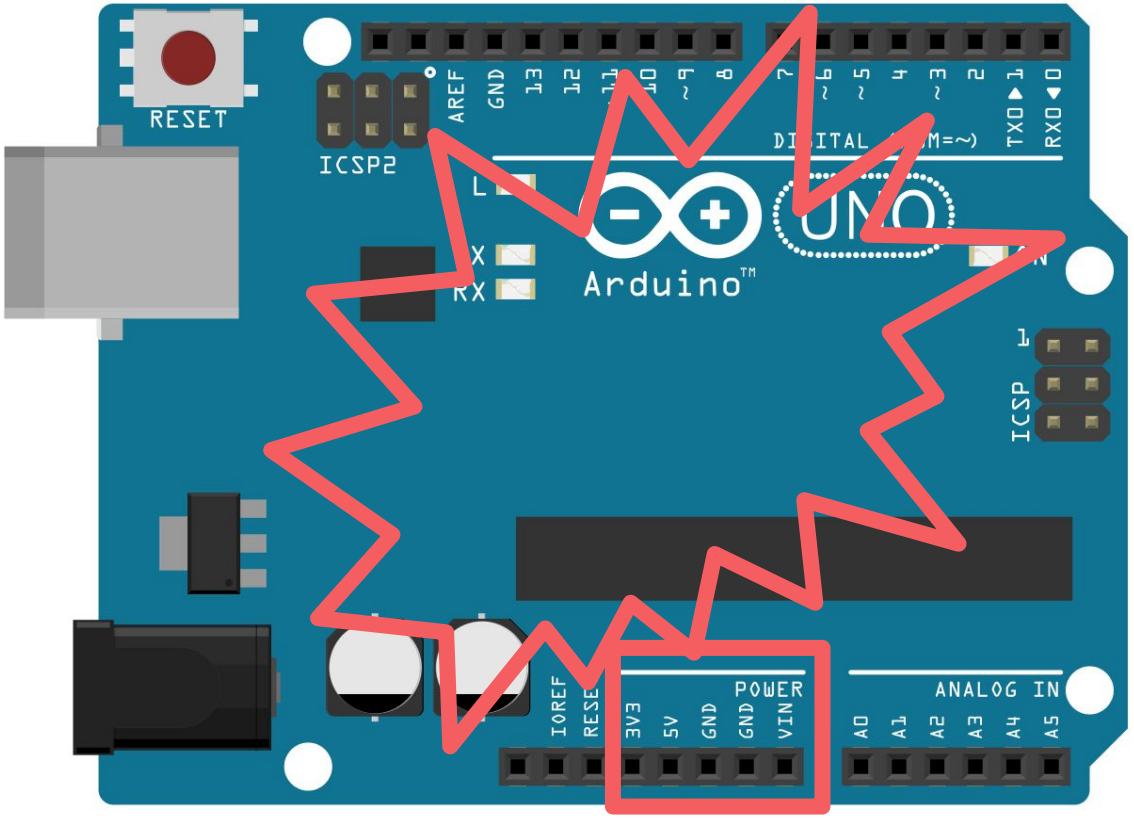
Alimentação

3v3

5v

GND

VIN - Ligado ao Conector de Alimentação, Pode ser usada para alimentar a placa.

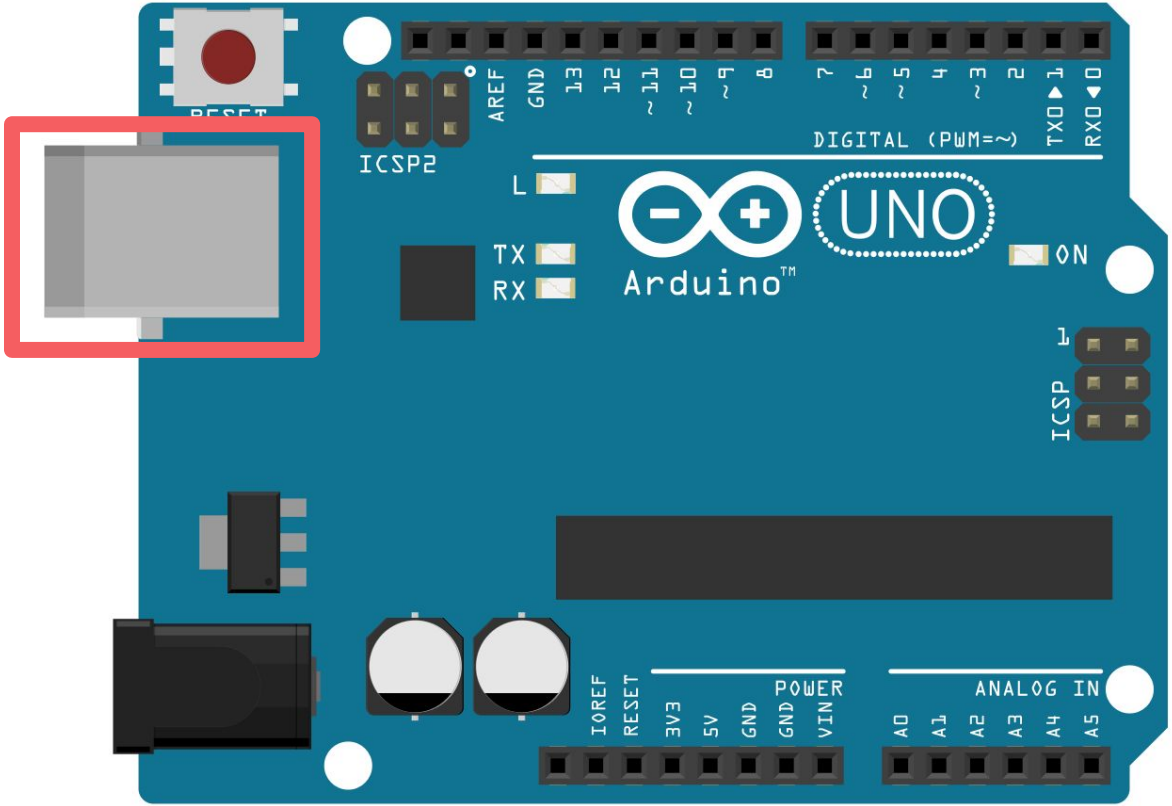


Cuidado ao Alimentar o Arduino, o limite da alimentação é de 6 - 20v.

Porém quando alimentado acima dos 12v uma carga consumindo mais de 600mA, do pino 5v, é capaz de superaquecer o regulador de tensão.

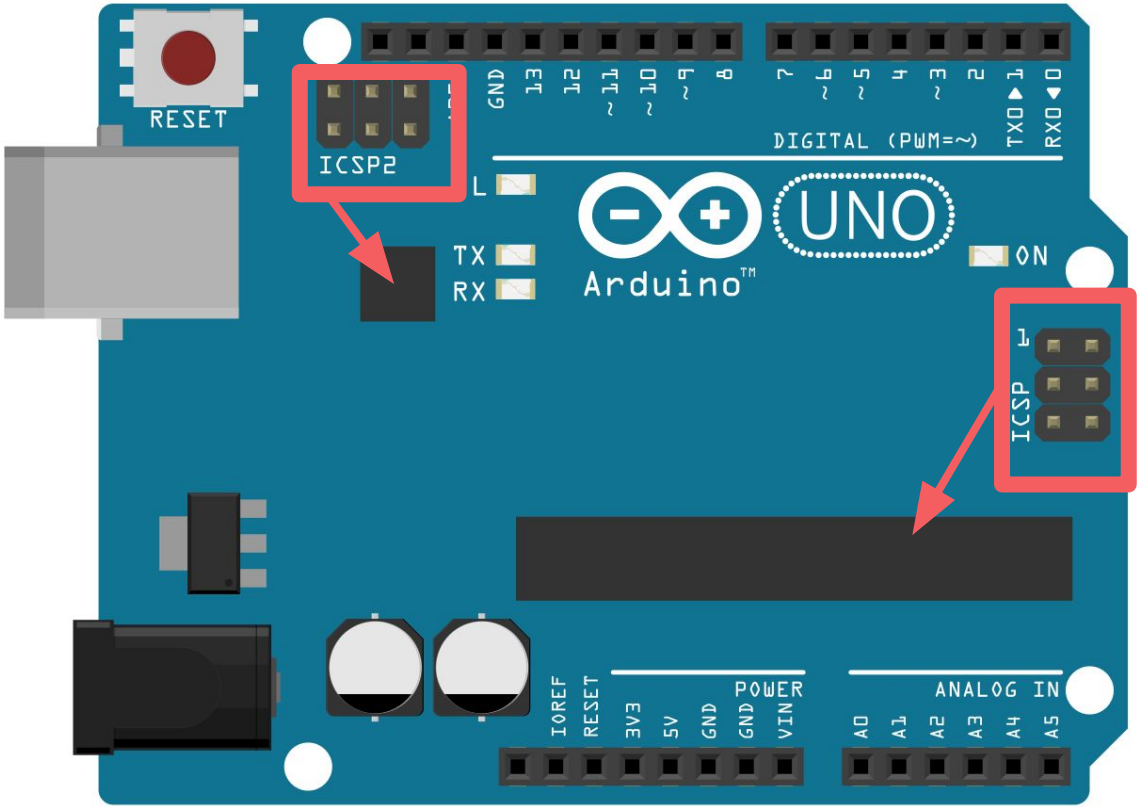
Alimentação

Na dúvida use uma fonte confiável conectada à entrada USB



Conector ICSP

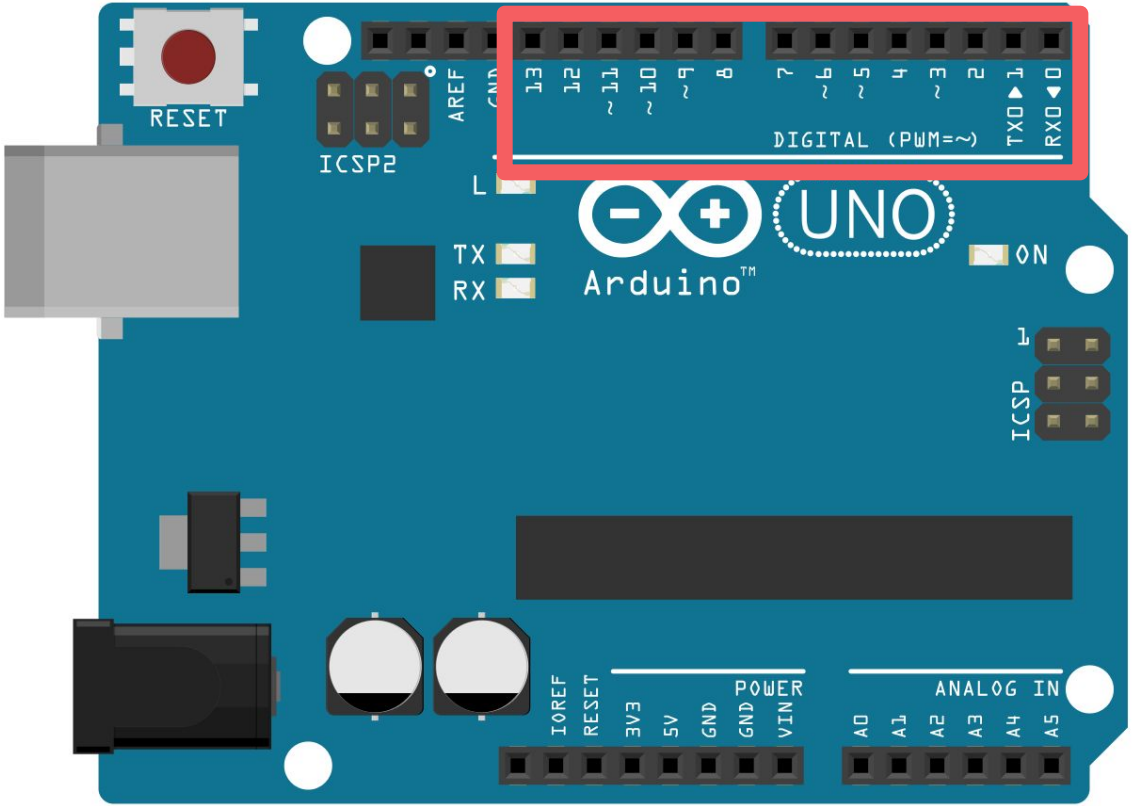
ICSP e um padrão de conexão serial, no arduino uno pode ser usado para gravar o bootloader



Conectores Digitais

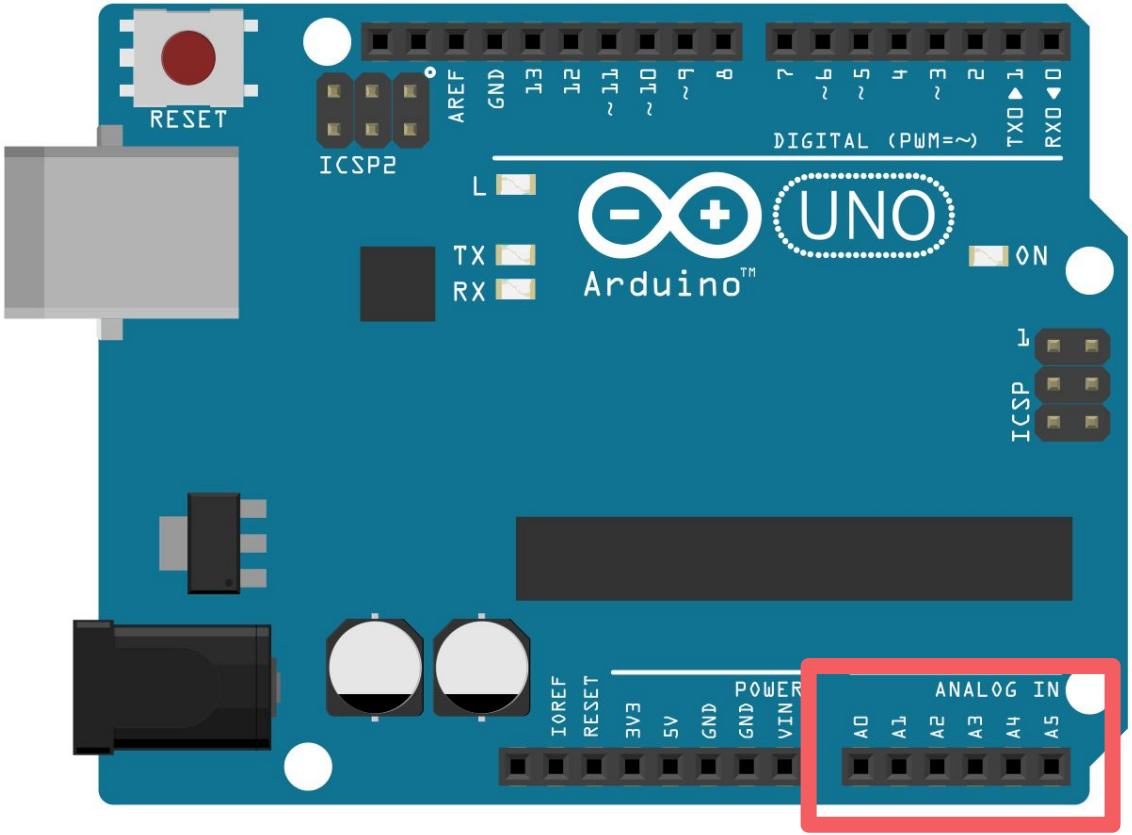
Os conectores digitais do arduino têm capacidade de entrada e saída.

*Os pinos marcados por [~] tem capacidade de saída PWM. Emulando uma saída analógica.



Conectores Analogicos

O Arduino possui 6 entradas analógicas identificadas de A0 a A5.

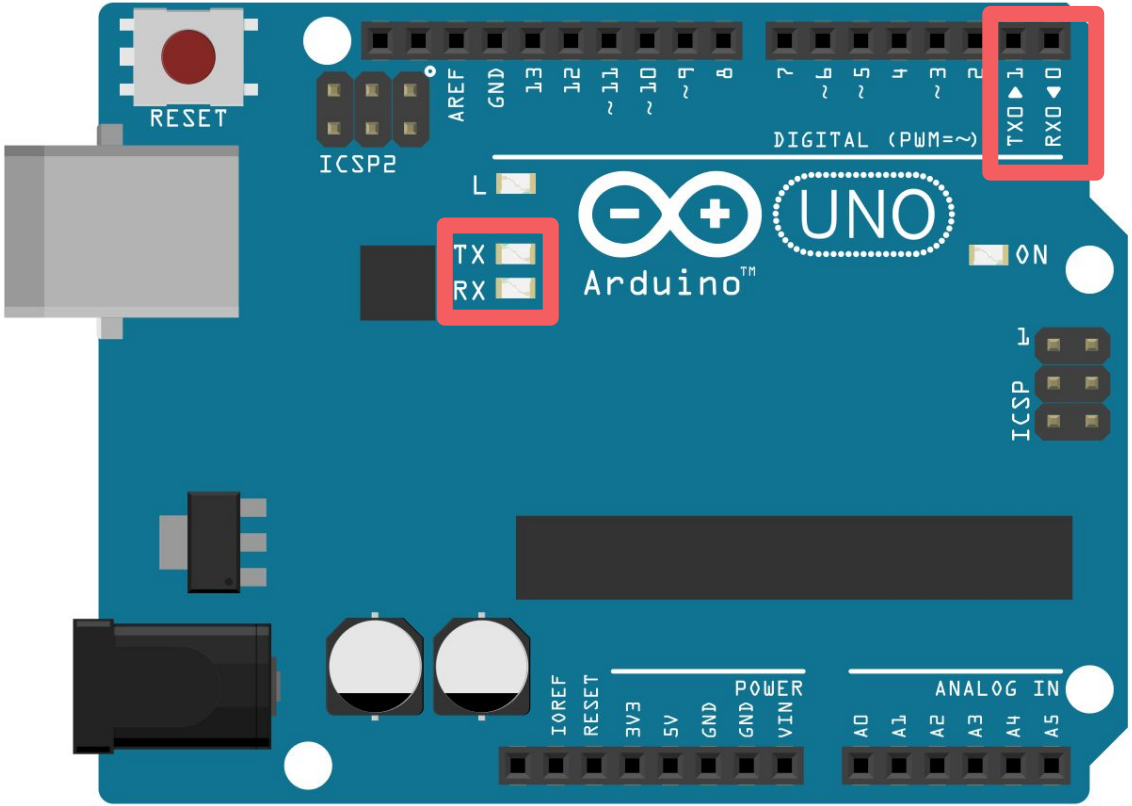


Conector e LEDs da porta serial

Os pinos digitais 0 e 1 são reservados para a comunicação serial.

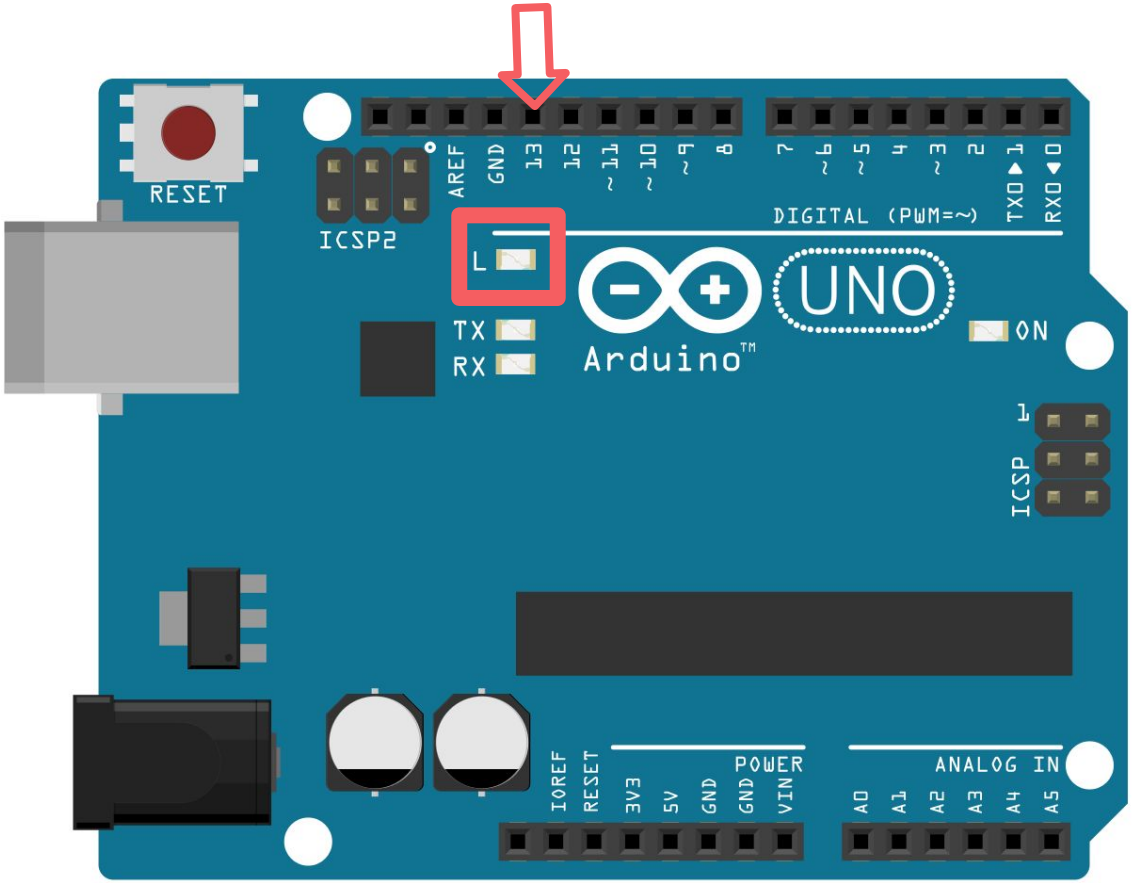
A atividade da porta serial é indicada por dois leds, RX (recepção pino 0) e TX (transmissão pino 1)

A comunicação serial no Arduino pode ser implementada tanto por hardware quanto por software.



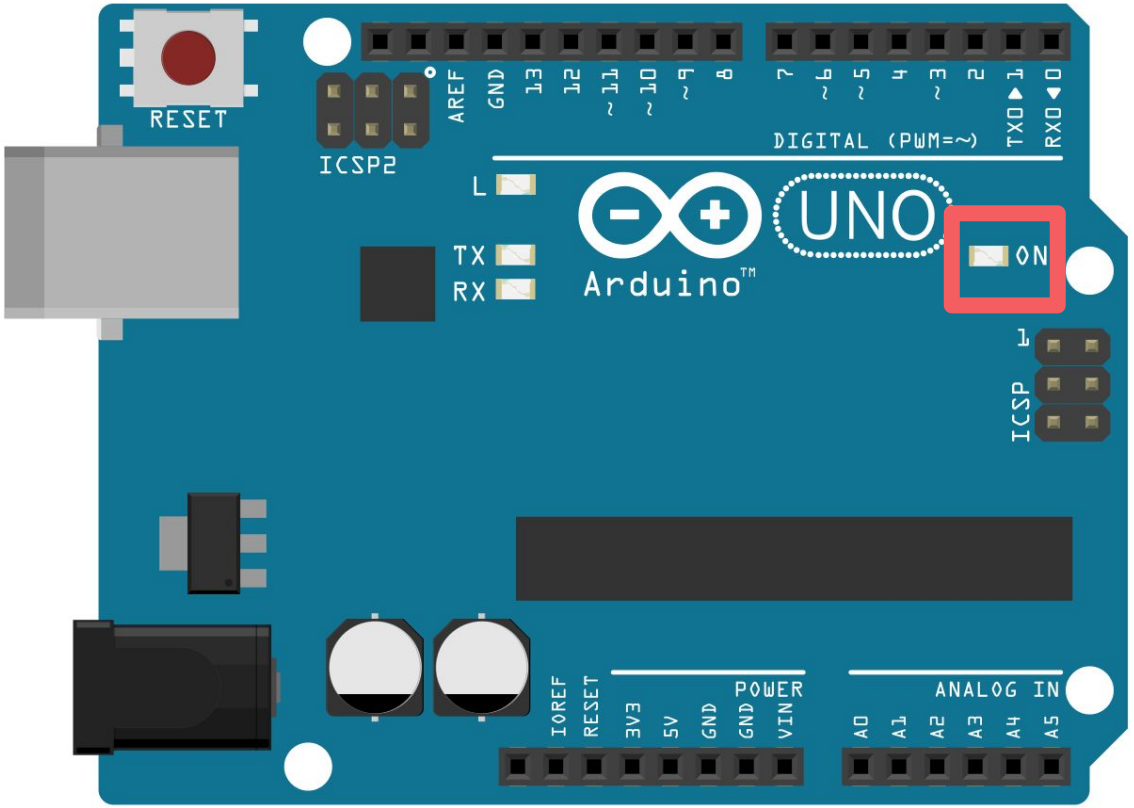
LED Interno

O LED Interno do Arduino está ligado ao pino digital 13



Power LED

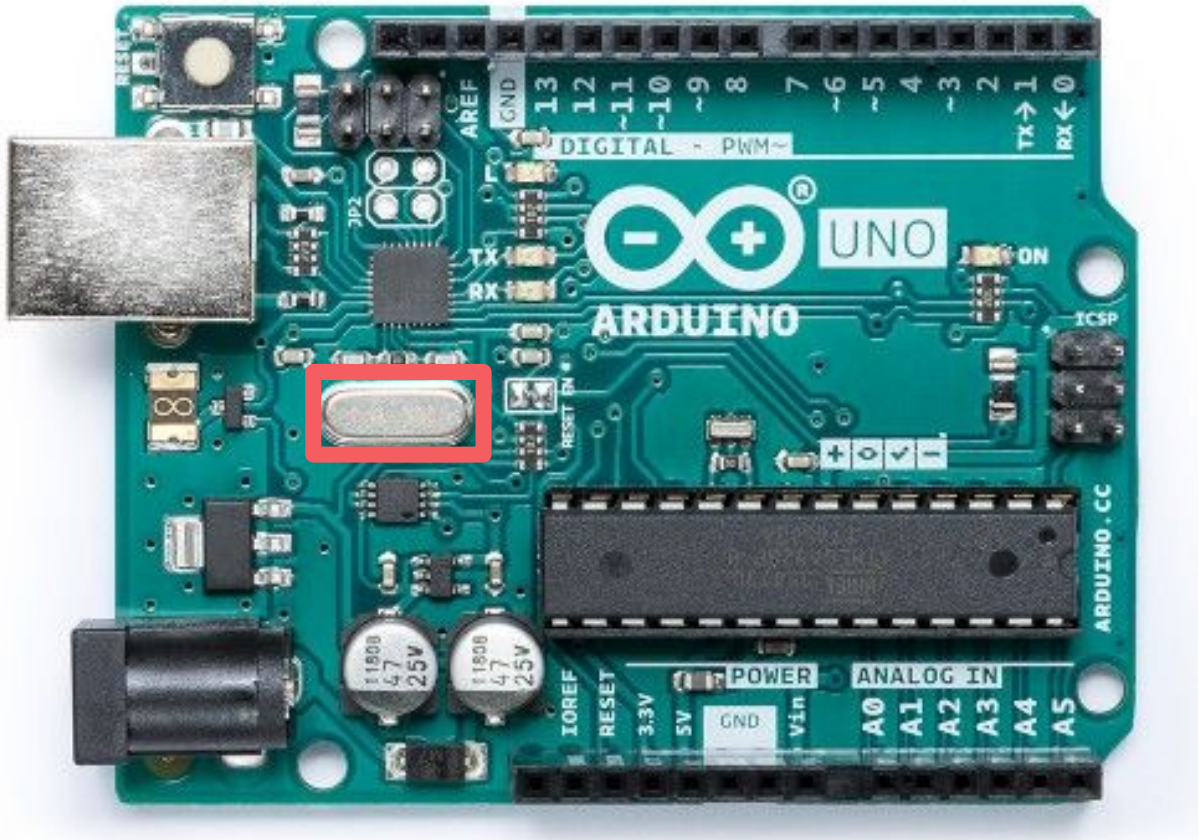
Indica que a placa está ligada.



Cristal Oscilador

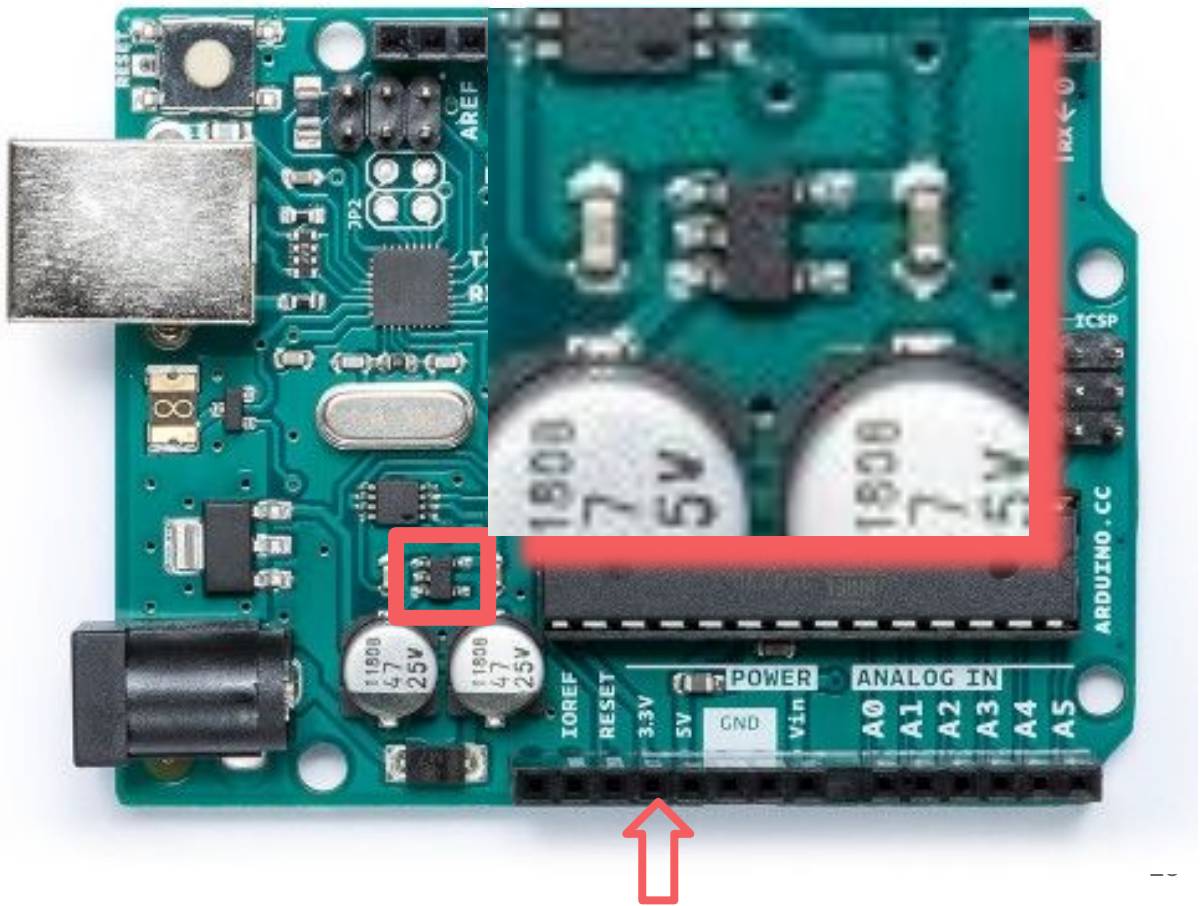
16Mhz - Controla a frequência de operação do micro controlador.

Alguns projetos substituem esse componente como forma de overclock ou downclock.



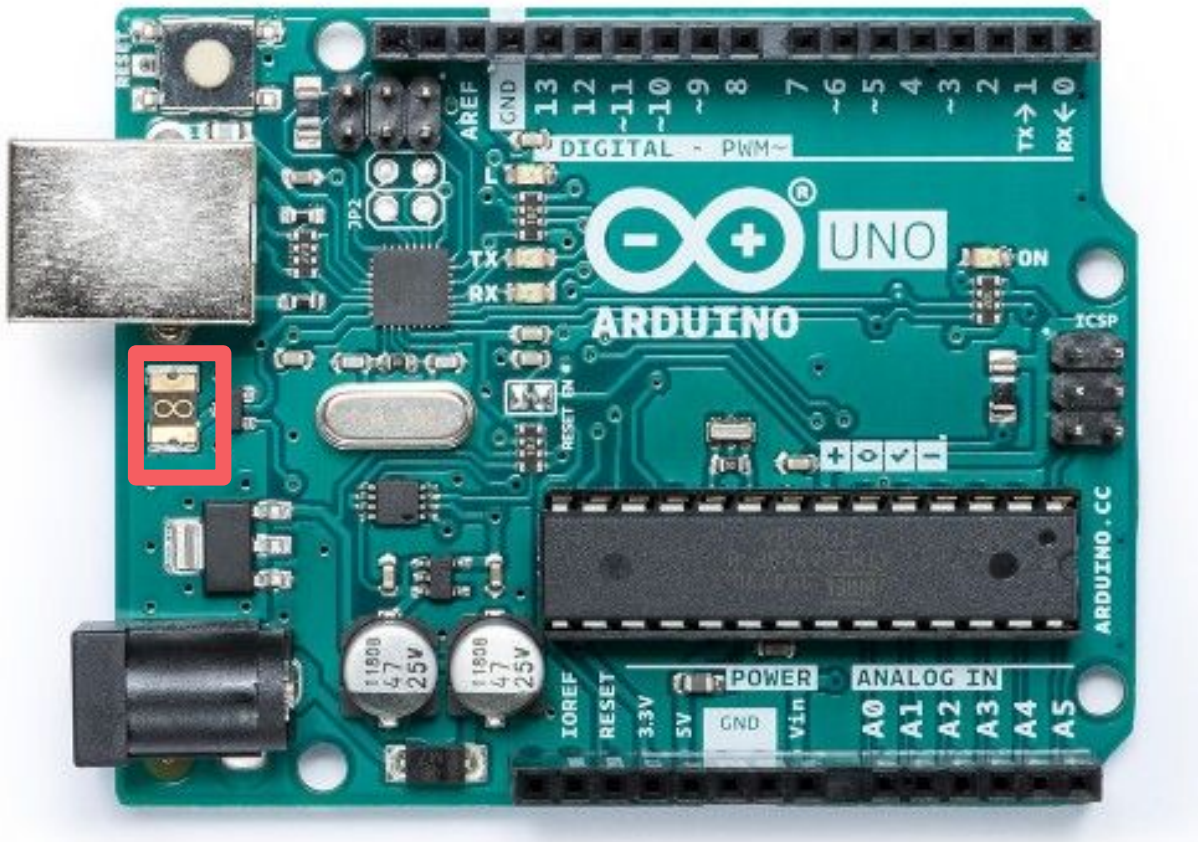
Regulador de Tensão 3.3v

http://www.ti.com/lit/ds/symlink/lp2985.pdf



Proteção

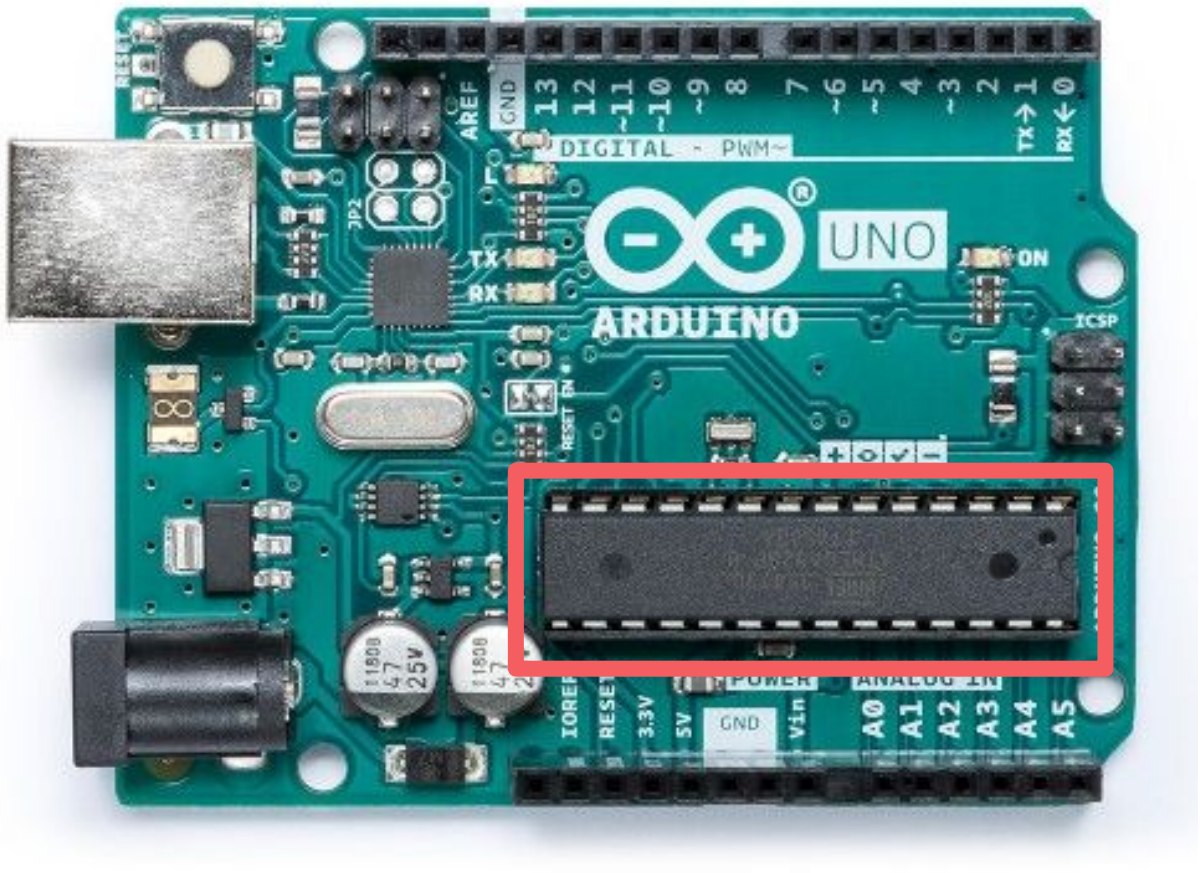
Funciona como um fusível, abrindo o circuito em caso de sobre alimentação



Microcontrolador

ATmega328/P

http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf



Microcontrolador

http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

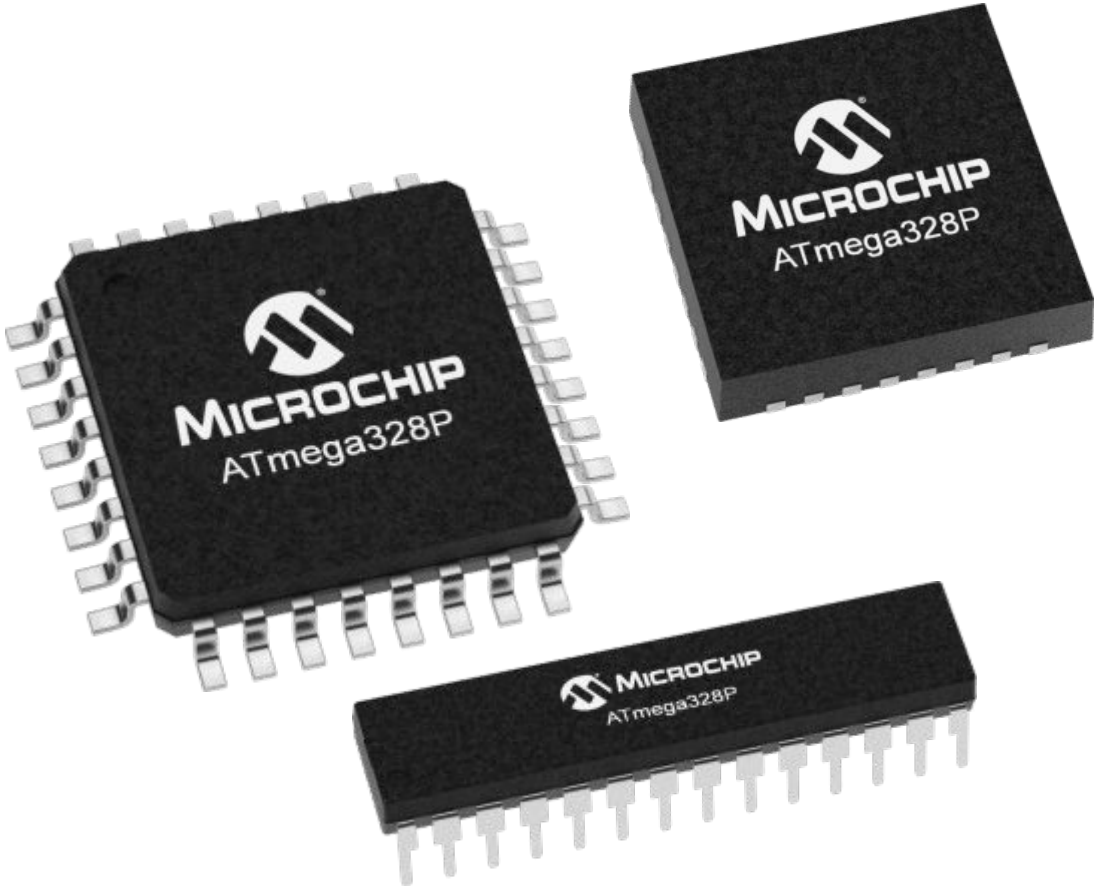
O Atmel® ATmega328 / P é um microcontrolador de 8 bits baseado na arquitetura RISC. Executa instruções em um único ciclo de clock, alcançando uma vazão próxima a 1MIPS por MHz.

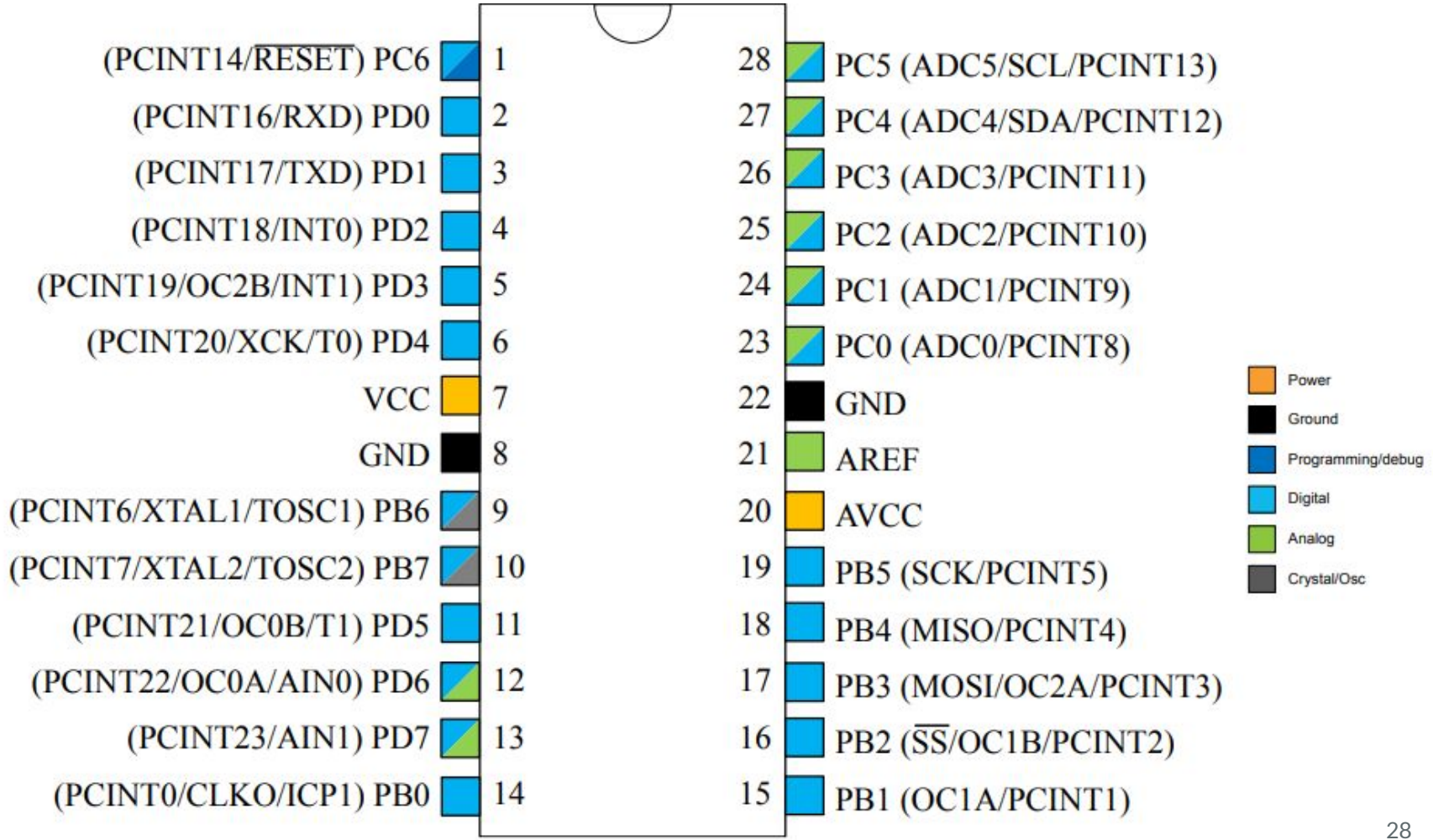
Microcontrolador

ATmega328/P

Memória

- Flash 32KB (Programa)
- SRAM 2KB (RAM)
- EEPROM 1Kb (ROM)



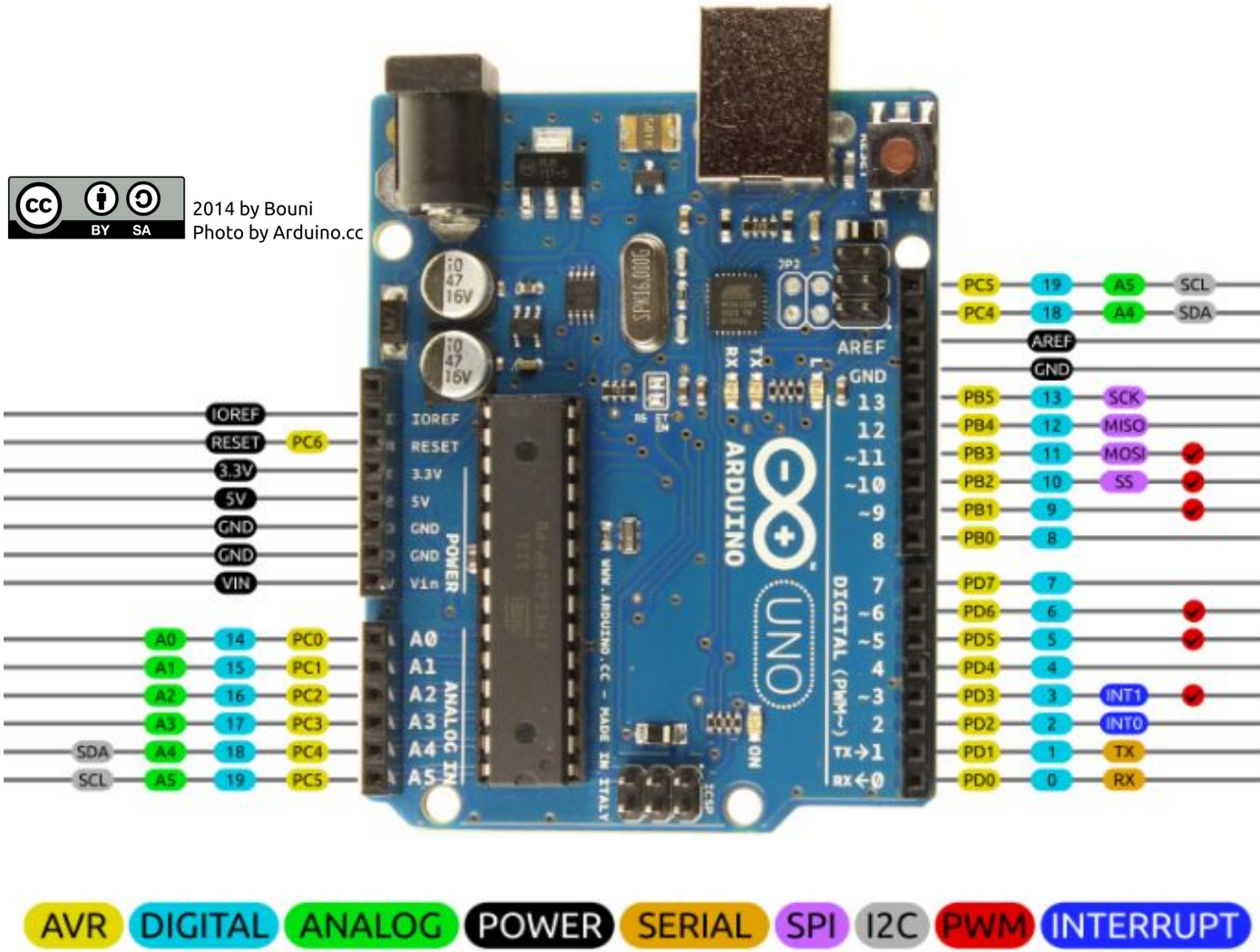

(PCINT14/RESET) PC6 1
(PCINT16/RXD) PD0 2
(PCINT17/TXD) PD1 3
(PCINT18/INT0) PD2 4
(PCINT19/OC2B/INT1) PD3 5
(PCINT20/XCK/T0) PD4 6
VCC 7
GND 8
(PCINT6/XTAL1/TOSC1) PB6 9
(PCINT7/XTAL2/TOSC2) PB7 10
(PCINT21/OC0B/T1) PD5 11
(PCINT22/OC0A/AIN0) PD6 12
(PCINT23/AIN1) PD7 13
(PCINT0/CLKO/ICP1) PB0 14
28 PC5 (ADC5/SCL/PCINT13)
27 PC4 (ADC4/SDA/PCINT12)
26 PC3 (ADC3/PCINT11)
25 PC2 (ADC2/PCINT10)
24 PC1 (ADC1/PCINT9)
23 PC0 (ADC0/PCINT8)
22 GND
21 AREF
20 AVCC
19 PB5 (SCK/PCINT5)
18 PB4 (MISO/PCINT4)
17 PB3 (MOSI/OC2A/PCINT3)
16 PB2 (SS/OC1B/PCINT2)
15 PB1 (OC1A/PCINT1)
Power
Ground
Programming/debug
Digital
Analog
Crystal/Osc

pinout

2014 by Bouni
Photo by Arduino.cc

https://github.com/Bouni/Arduino-Pinout

- Arduino Uno
https://github.com/Bouni/Arduino-Pinout/blob/master/Arduino%20Uno%20R3%20Pinout.png



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

Simulando o Arduino no Tinkercad

https://www.tinkercad.com

- Criando uma conta
- Criando um novo projeto
- Inserindo Componentes
- Inserindo Código
- As funções setup() e loop().

Simulando o Arduino no Tinkercad

https://www.tinkercad.com

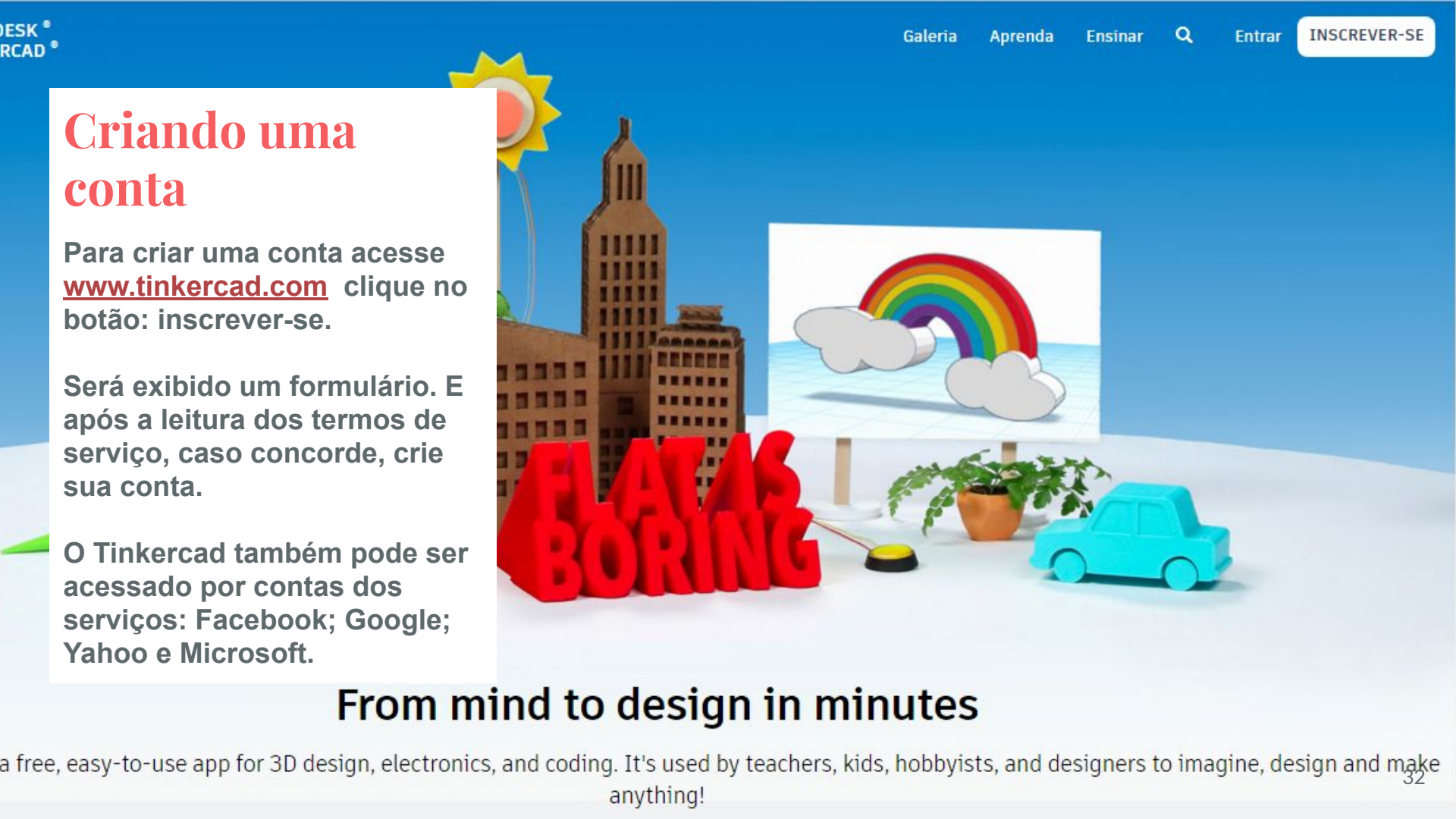
- A Plataforma online Tinkercad, da Autodesk, oferece uma alternativa simples para a simulação de pequenos projetos eletrônicos.
- Também suporta: Modelagem 3d e Programação por blocos.

Criando uma conta

Para criar uma conta acesse www.tinkercad.com clique no botão: inscrever-se.

Será exibido um formulário. E após a leitura dos termos de serviço, caso concorde, crie sua conta.

O Tinkercad também pode ser acessado por contas dos serviços: Facebook; Google; Yahoo e Microsoft.

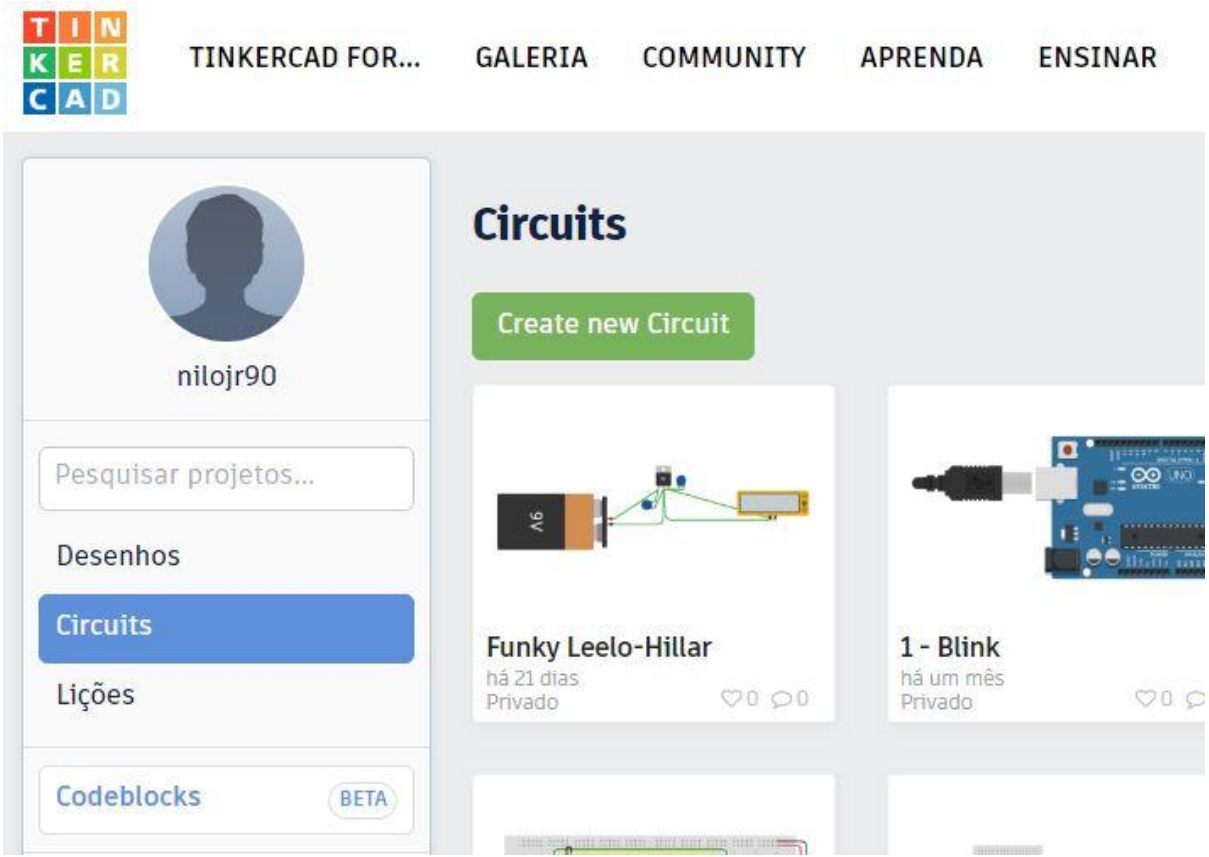


Criando um novo projeto

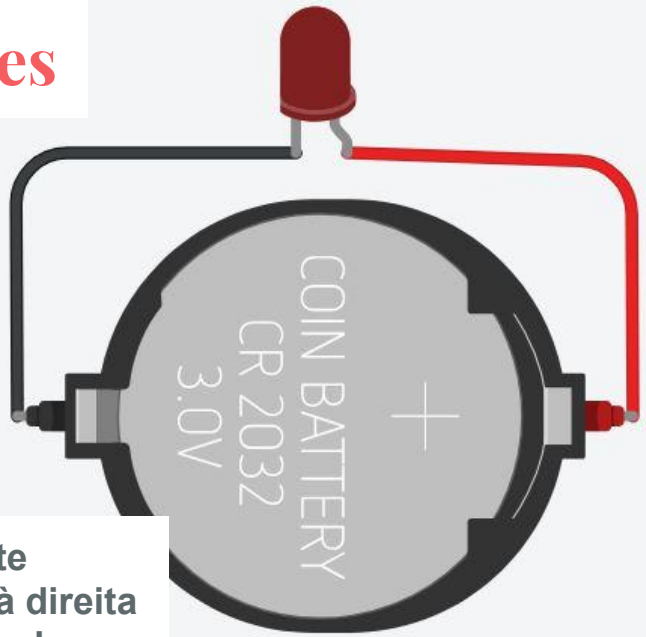
Em seu "Dashboard".

Escolha a opção "Circuits", no painel à esquerda.

Então
escolha a opção "Create new Circuit".

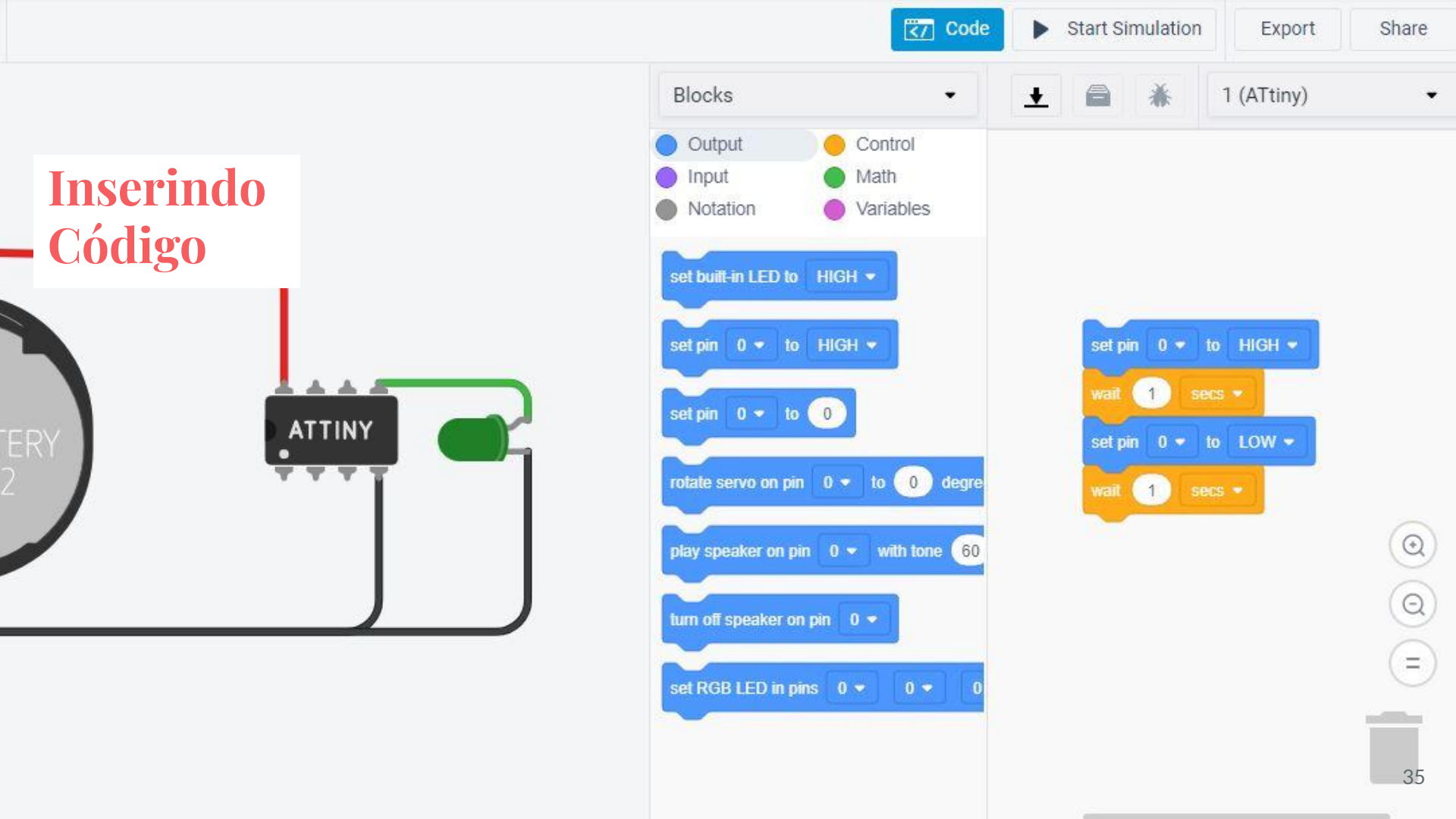


Inserindo Componentes

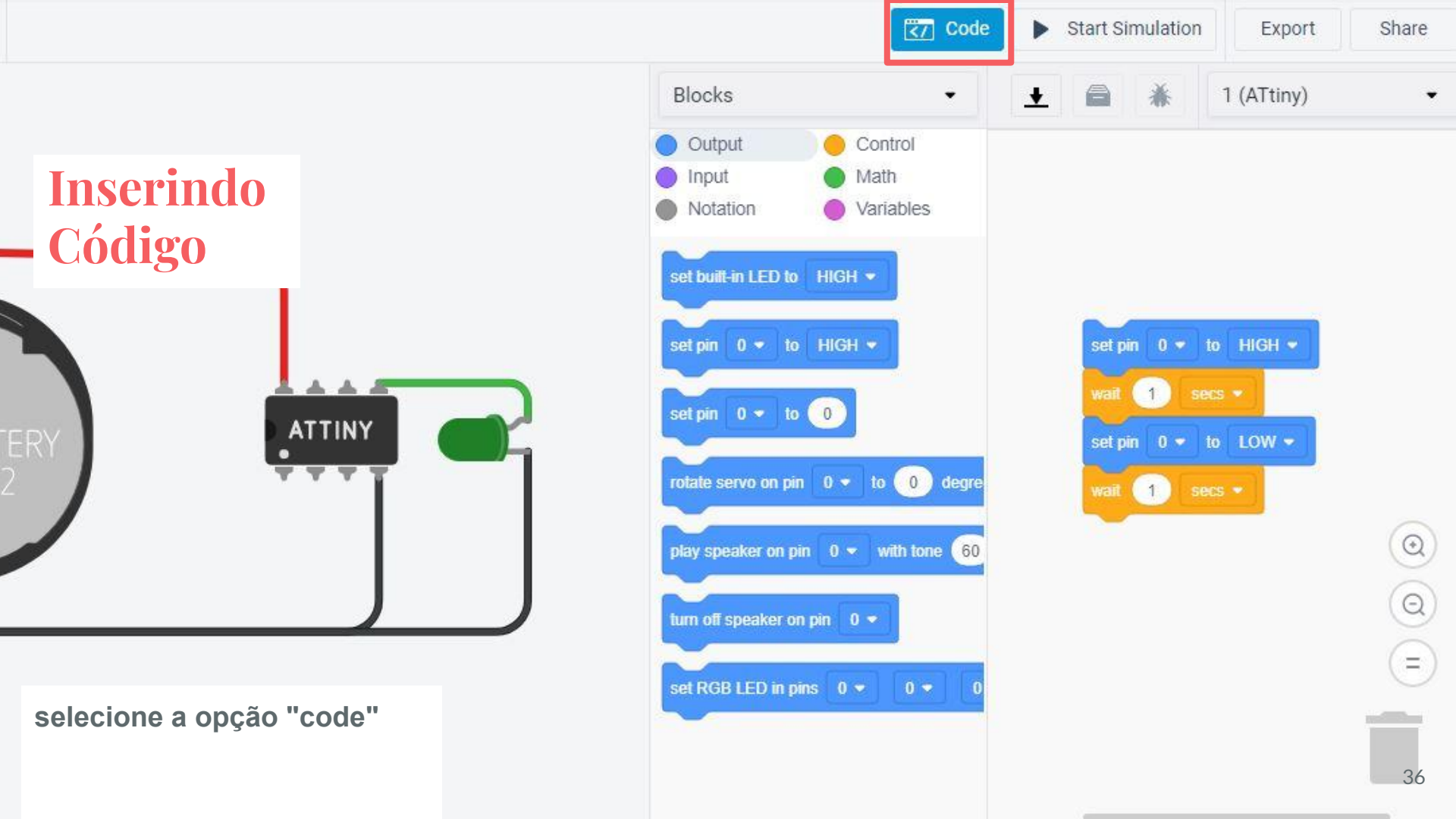


Arraste o componente desejado da secção à direita e solte-o em sua área de trabalho.

Inserindo Código

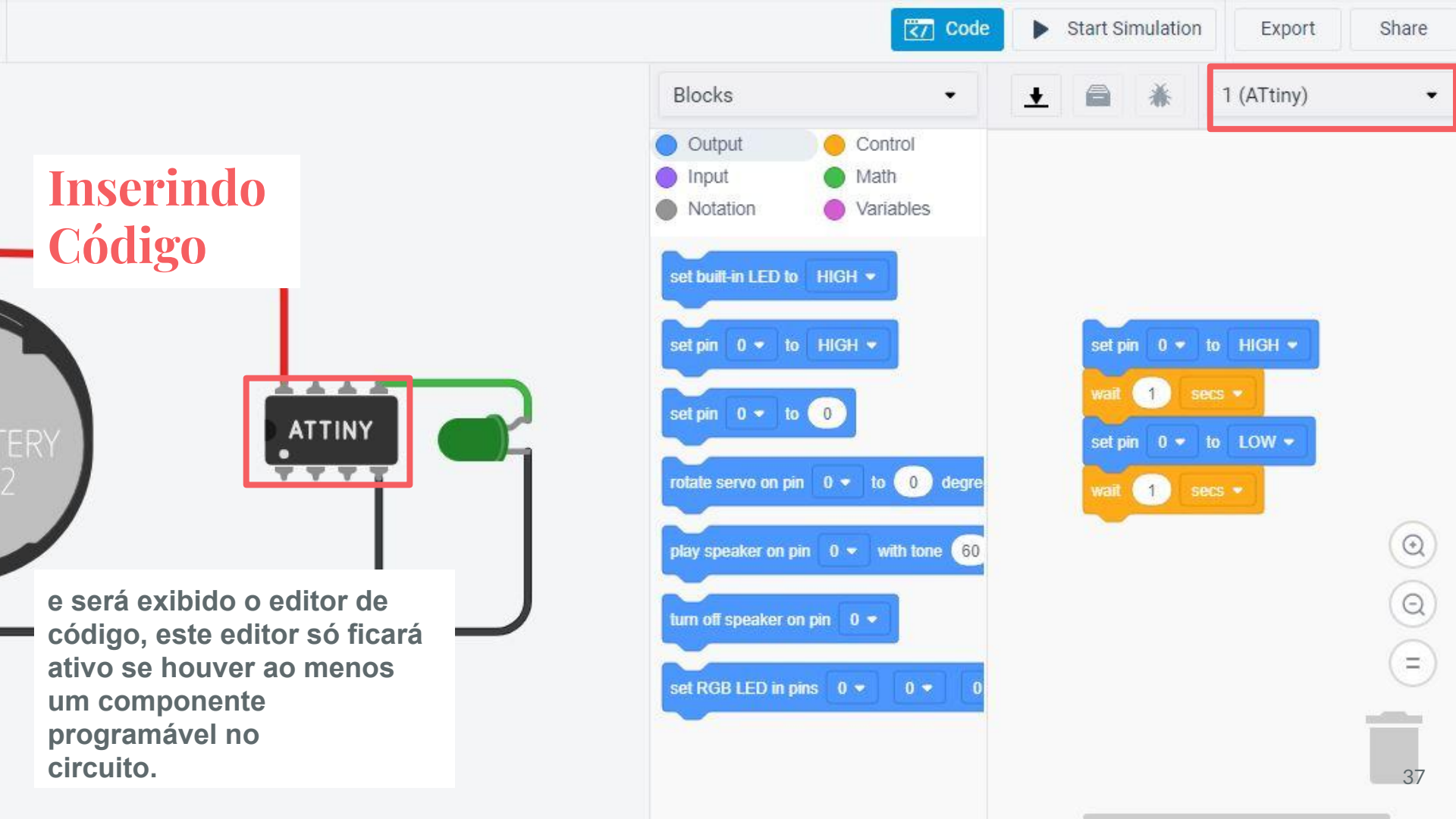


Inserindo Código



selecione a opção "code"

Inserindo Código



e será exibido o editor de código, este editor só ficará ativo se houver ao menos um componente programável no circuito.

Inserindo Código

e possível editar seu programa de três modos: Blocks; Blocks + Text ou Text.

`setup()`

- A função ‘`setup()`’ é chamada automaticamente pelo bootloader do microcontrolador assim que é ligado ou reiniciado.
- E executada apenas uma vez, normalmente usada para a configuração das entradas, saídas e definição das variáveis usadas no programa.
- A função ’`setup()`’ pode ser considerada o cabeçalho do programa.

https://www.arduino.cc/reference/en/language/structure/sketch/setup/

Loop ( )

- Já a função ‘loop()’, inicia sua execução logo após o termino da função ’setup()’.
- E executada repetidamente enquanto o microcontrolador estiver ligado.
- A função ‘loop()’pode ser considerada o corpo do programa.

https://www.arduino.cc/reference/en/language/structure/sketch/loop/

```
// Executado apenas uma vez.
void setup(){
  // O pino '0' ZERO será usado como saída.
  pinMode(0, OUTPUT);
}

// Executado repetidas vezes.
void loop(){
  //Define a tensão como ALTA no pino '0' ZERO
  digitalWrite(0, HIGH);
  // Espera por 1000 milissegundos)
  delay(1000);
  //Define a tensão como BAIXA no pino '0' ZERO
  digitalWrite(0, LOW);
  // Espera por 1000 milissegundos)
  delay(1000);
}
```

A linguagem do Arduino

https://www.arduino.cc/reference/en/

http://ordemnatural.com.br/pdf-files/CartilhadoArduino_ed1.pdf

https://multilogica-shop.com/Referencia
https://github.com/liffiton/Arduino-Cheat-Sheet/blob/master/Arduino%20Cheat%20Sheet.pdf

A linguagem de programação Arduino pode ser dividida em três partes principais:

- funções.
- valores.
- estrutura.

Funções

millis()

- Retorna o número de milissegundos desde que a placa do Arduino começou a executar o programa atual.
- `time = millis(); //unsigned long`

https://www.arduino.cc/reference/en/language/functions/time/millis/

```
unsigned long tempo;

void setup(){
  Serial.begin(9600);
}
void loop(){
  Serial.print("Time: ");
  tempo = millis();

  Serial.println(tempo);//imprime o tempo desde que o
programa iniciou
  delay(1000);          //espera 1 segundo
}
```

Variáveis

tipos*

- String
- boolean
- byte
- double
- float
- int
- long
- short

declaração

- String
- boolean
- byte
- double
- float
- int
- long
- short

```
int pinoLED = 5;         // LED no pino 5
int pinoBotao = 10;    // botão com retorno
por mola: 10_/_GND

bool ligado = false;

void setup()
{
  pinMode(LEDpin, OUTPUT);
  pinMode(switchPin, INPUT);
}
```

constantes

```
// definição de constantes
const int PINO_LED = 5;
const float ACELERACAO = 9.8;

// constantes padrão da linguagem

HIGH            | LOW
INPUT           | OUTPUT
INPUT_PULLUP    | LED_BUILTIN

true            | false
```

Estutura

controle

- break
- continue
- do...while
- else
- for
- goto
- if...else
- return
- switch...case
- while

for

```
for (int i=0; i <= 255; i++){
    analogWrite(10, i);
    delay(10);
    }
```

operadores

Comparação

- !=
- <
- <=
- ==
- >
- >=

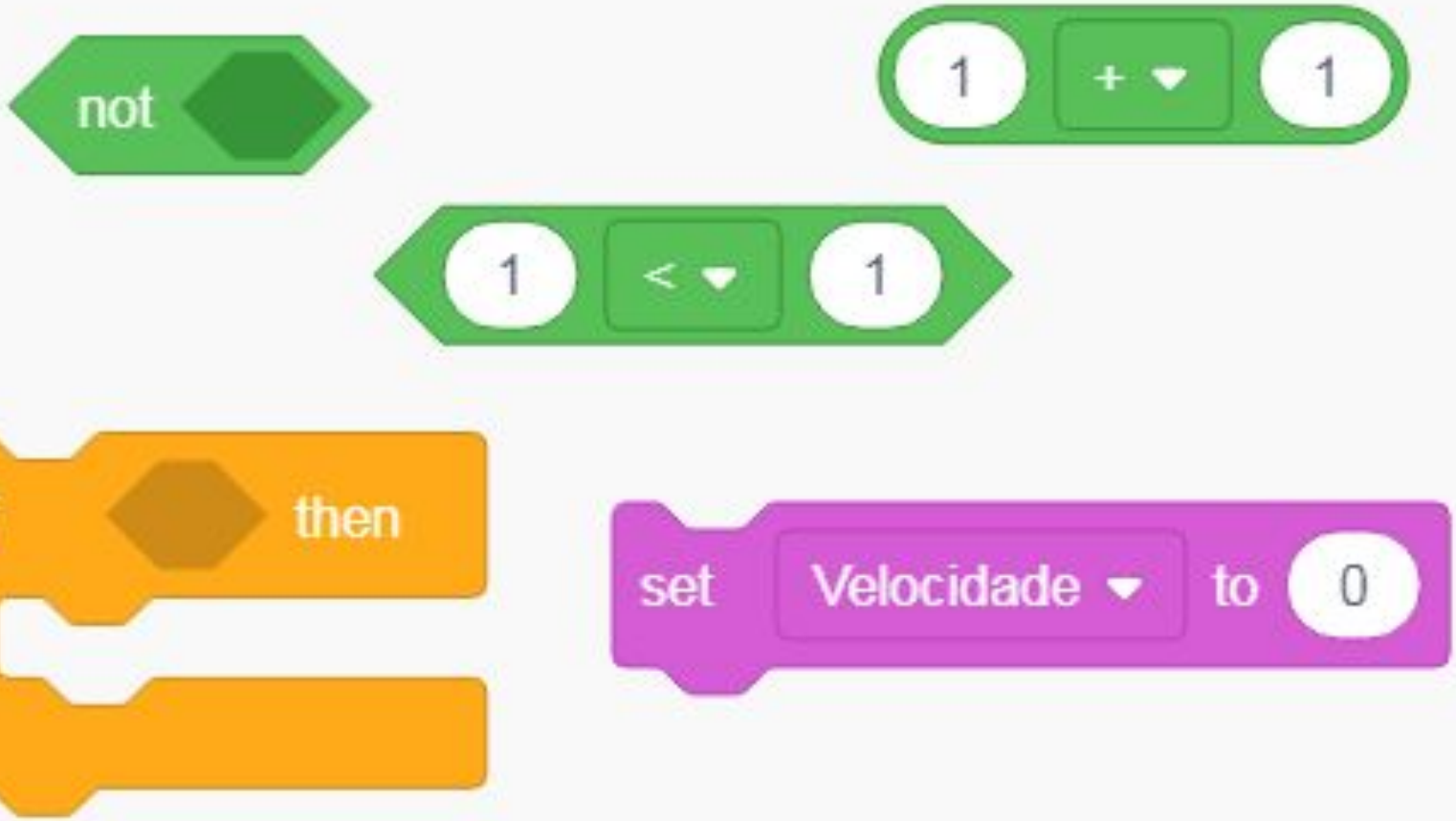
Booleanos

- ! (negação)
- && (e)
- || (ou)

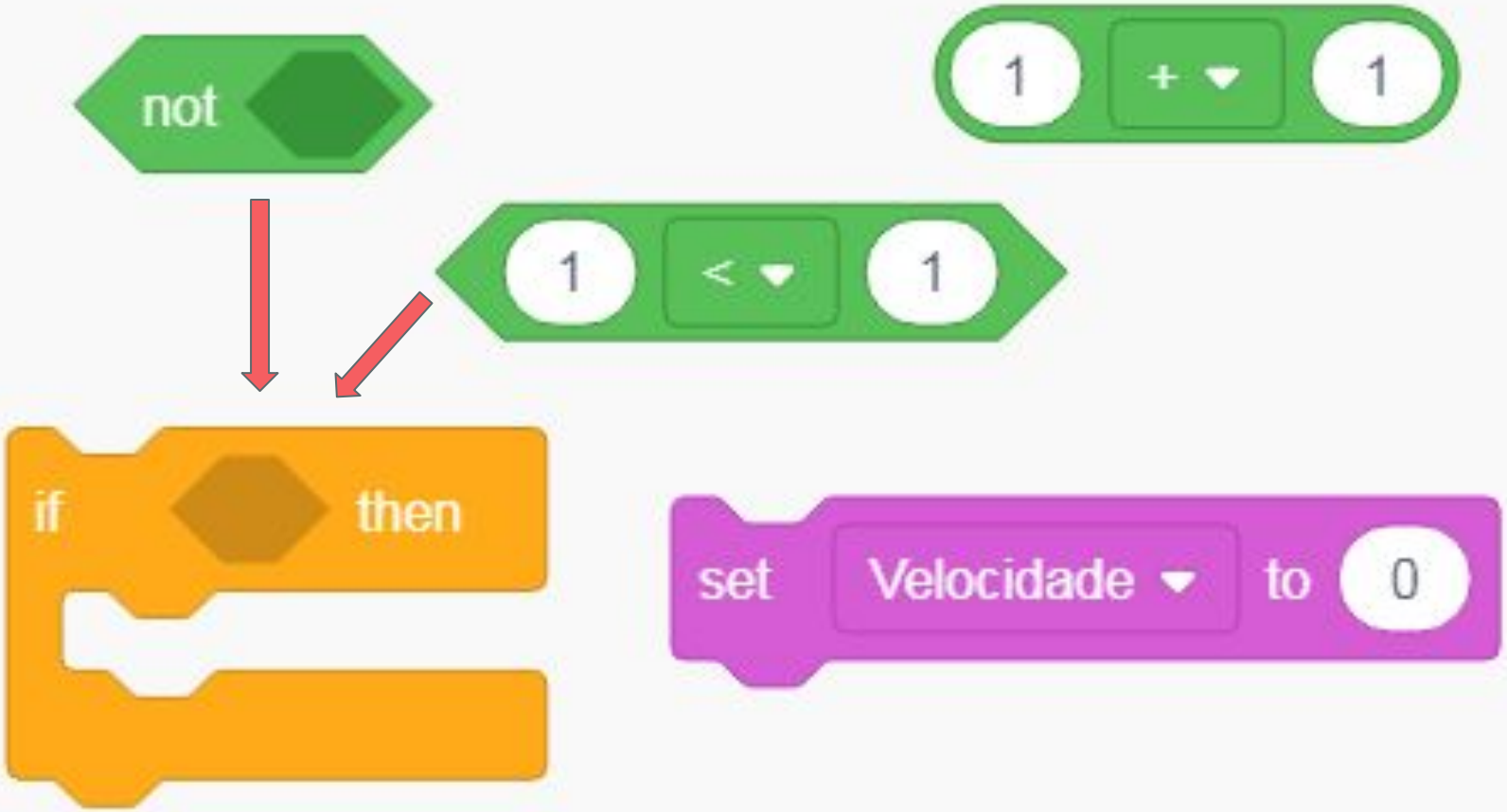
Aritméticos

- % (resto)
- *
- +
- -
- /
- = (atribuição)

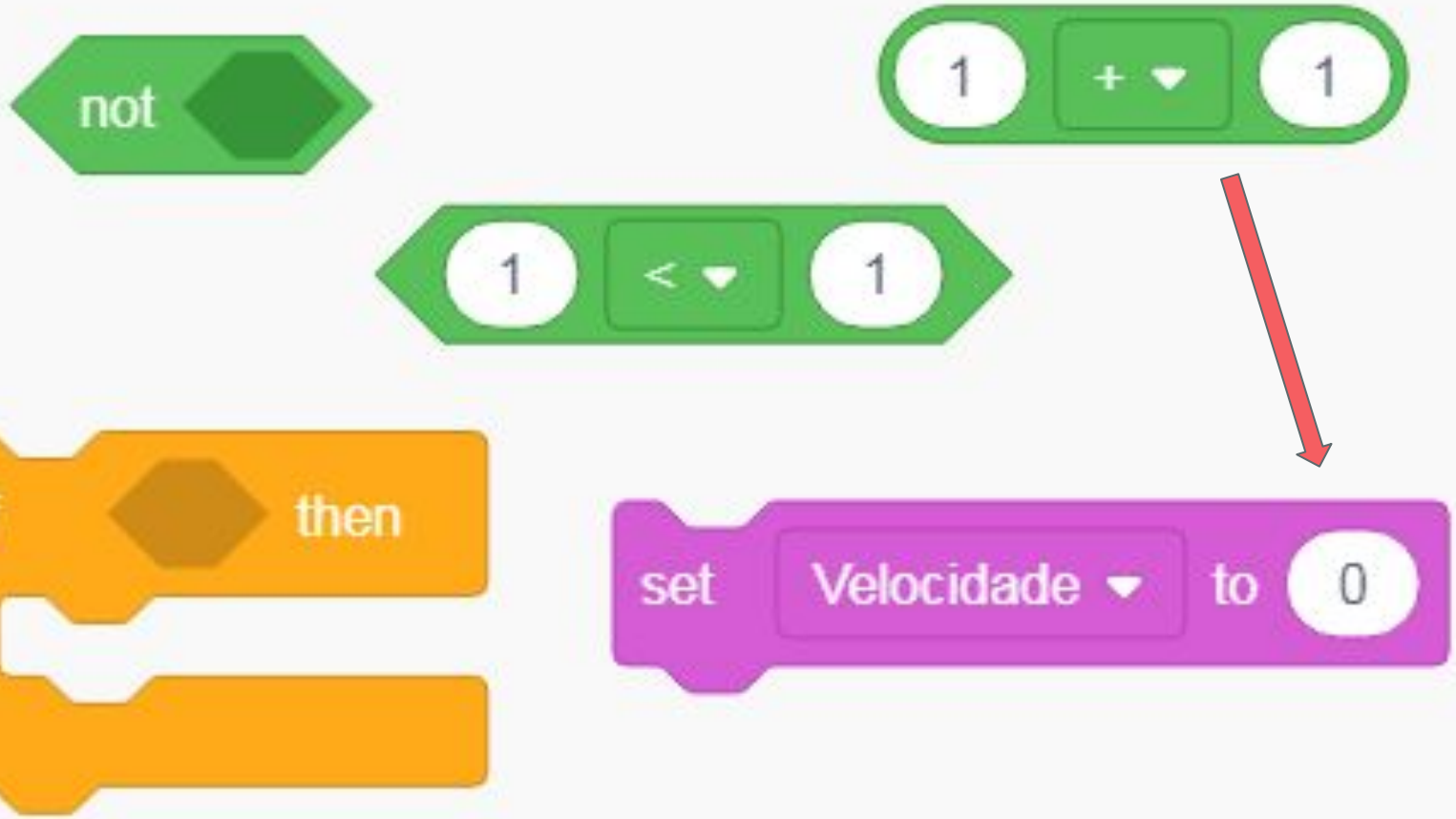
operadores



operadores



operadores



Entrada e Saída

https://www.arduino.cc/reference/en/

- A Plataforma online Tinkercad, da Autodesk, oferece uma alternativa simples para a simulação de pequenos projetos eletrônicos.
- Também suporta: Modelagem 3d e Programação por blocos.

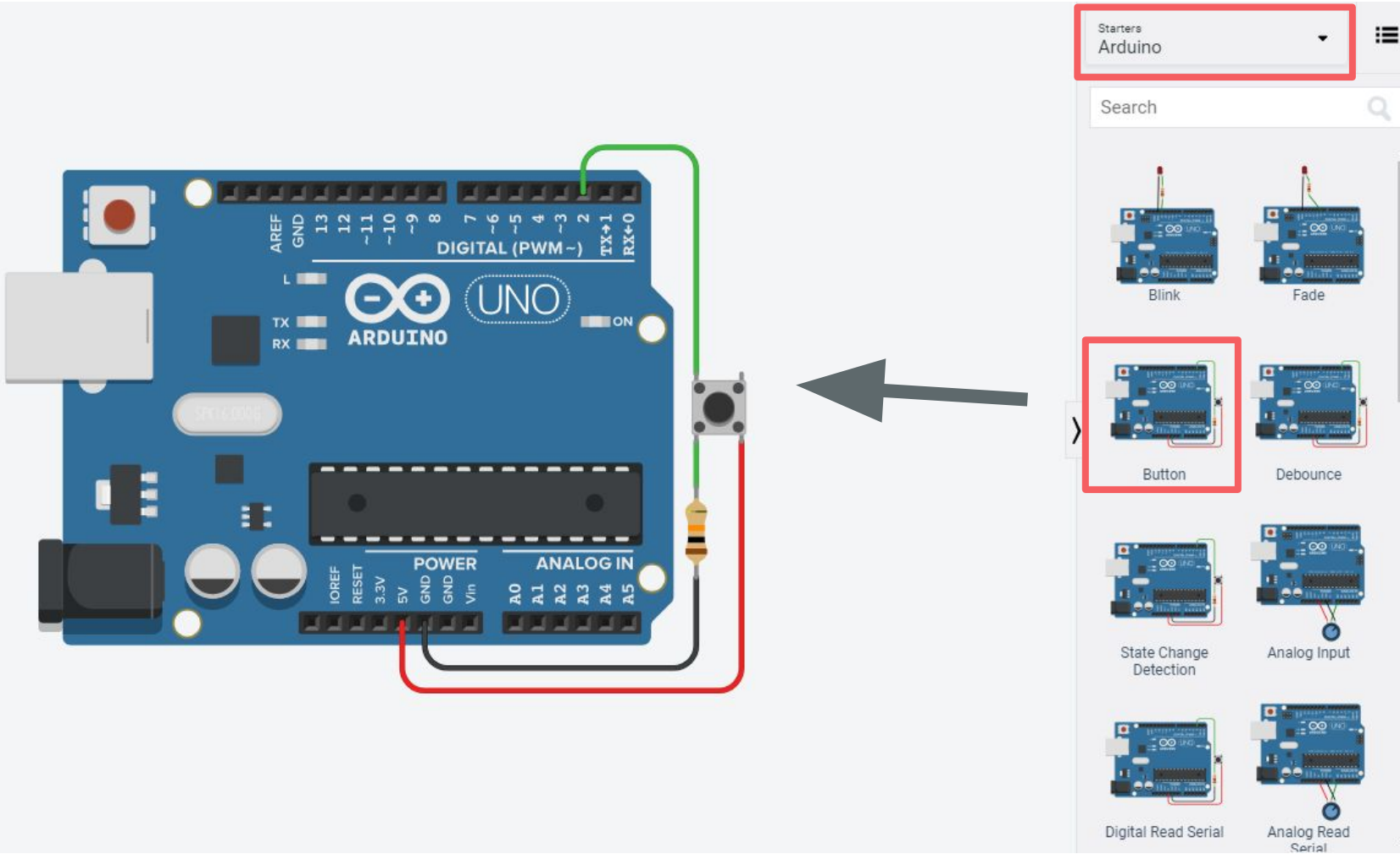
Entrada e Saída Digital

E/S Digital

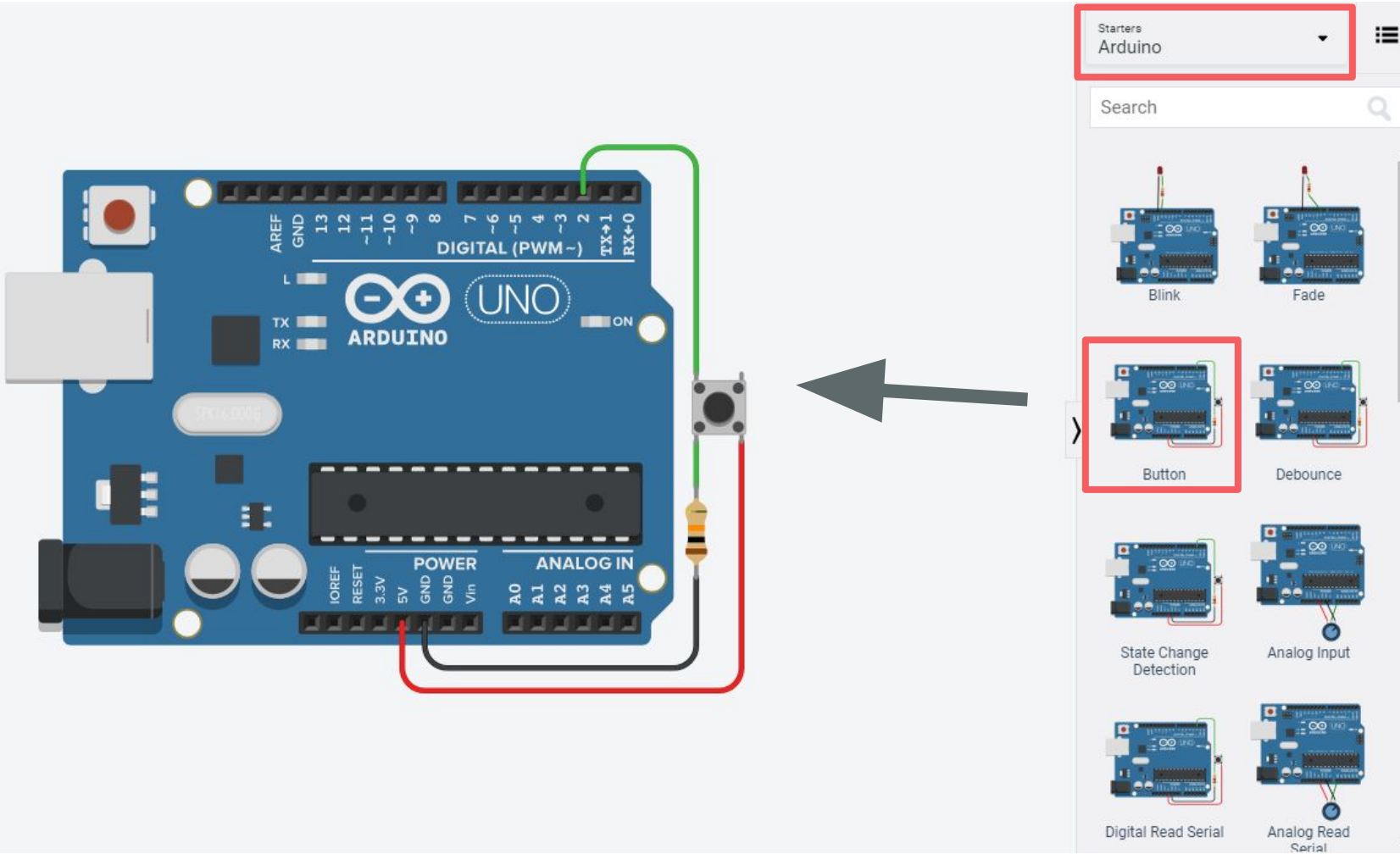
- pinMode()
- digitalRead()
- digitalWrite()

```
/* pinMode(pino,modo) - Configura o pino
  pino - pino a ser configurado
  modo - modo de uso pode ser:
  INPUT, OUTPUT, ou INPUT_PULLUP***
  ***ainda não pode ser simulado no
tinkercad

pinMode(13,OUTPUT);

digitalRead() // Starter Arduino Button
digitalWrite() // Starter Arduino Blink
```

Starter Arduino Button



Starter Arduino Button



Entrada e Saída Analógica

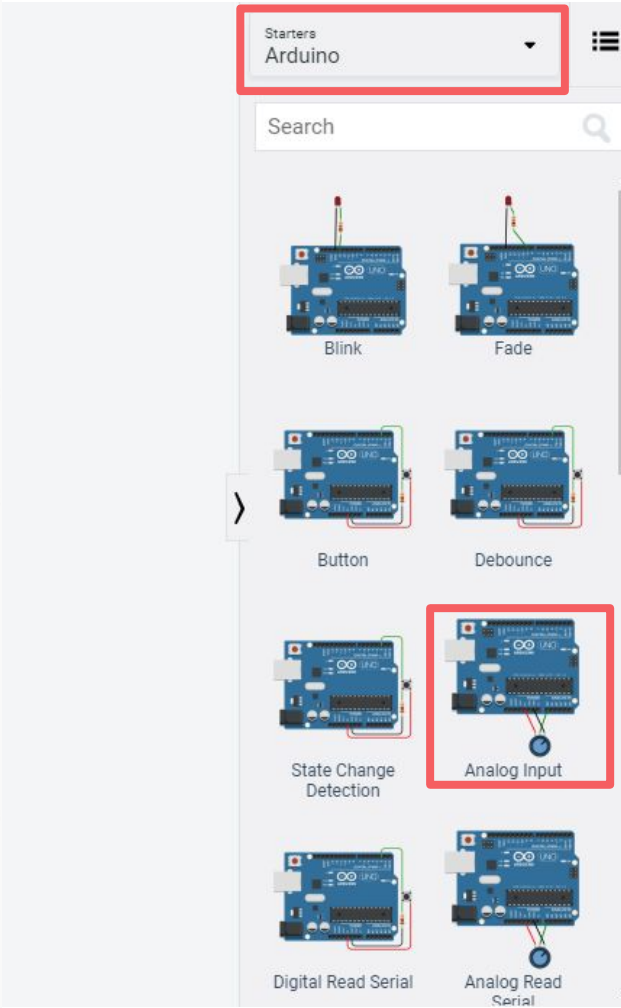
E/S Analógica

- analogRead()
- analogWrite()
- analogReference()

```
analogRead(0); // lê um valor
analógico,nesse caso  do pino A0, os
pinos de A0 a A5 são sempre entradas.

//analogRead() - retorna um inteiro de 0
a 1023, dependendo da tensão lida 0v = 0,
5v = 1023.

// Starter Arduino Analog Input
```

Starter Analog Input

```
comment  read the value from the sensor
set  sensorValue  to  read analog pin  A0
comment  turn the LED on
set built-in LED to  HIGH
comment  stop the program for the <sensorValue> millise…
wait  sensorValue  milliseconds
comment  turn the LED off
set built-in LED to  LOW
comment  stop the program for the <sensorValue> millise…
wait  sensorValue  milliseconds
```



Comunicação Serial

*código morse

- M.1677 : International Morse code

(Sinal de Início) -.-.-

. .-. -... .- ... .

(Fim da Transmissão) . - . - .[+]- . -[k]

http://www.itu.int/rec/R-REC-M.1677-1-200910-I/

- M.1677 : International Morse code

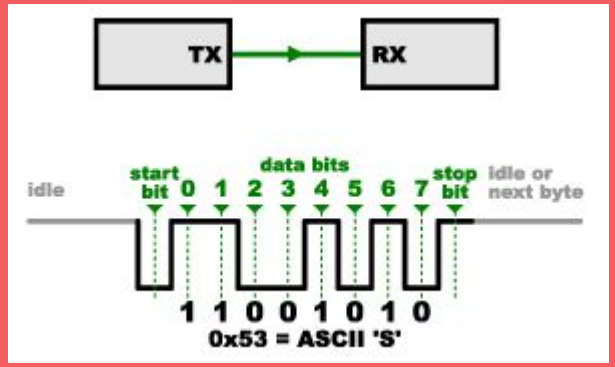
(Sinal de Início) -.-.-

.[e] .-.[r] -...[b] .-[a] ...[s] .[e]

(Fim da Transmissão) . - . - .[+]- . -[k]

http://www.itu.int/rec/R-REC-M.1677-1-200910-I/

serial assíncrono



0[1100][1010]1 > 01010011 ascii(**S**)

https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi

serial

begin()

pirntln()

write()

read()

end()

Serial.available()

```
void setup() {
    Serial.begin(9600);
// abre a porta serial e define a velocidade
para 9600 bps

}

void loop() {
  Serial.print(33);

  Serial.println(01010011, BIN); //S

  Serial.write(33); //00100001

  Serial.read();
```

serial

begin()

pirntln()

write()

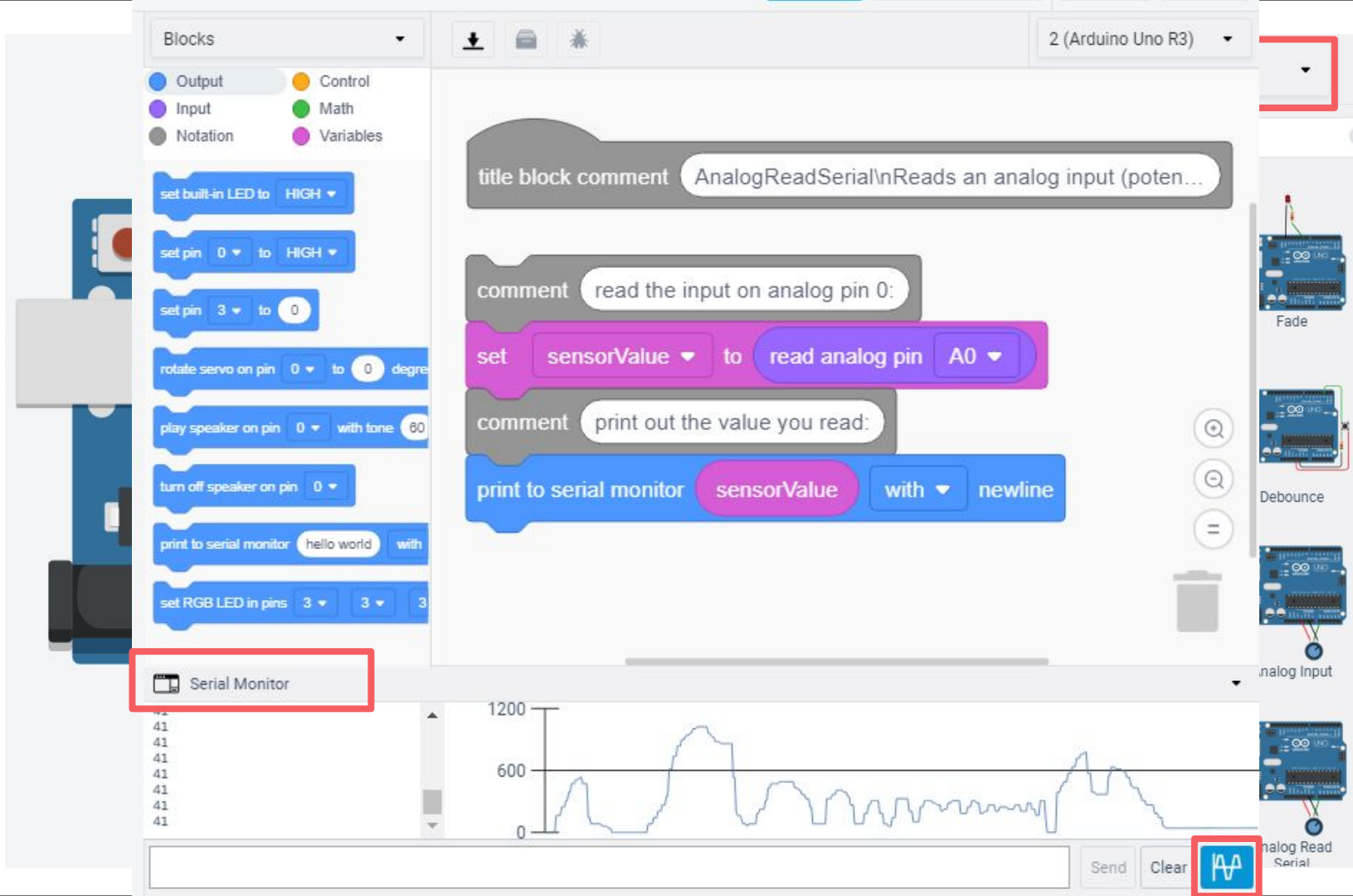
read()

end()

Serial.available()

```
Serial.end(); //desliga a porta serial.

Serial.available(); //S

Serial.write(33); //00100001

Serial.read();

//Starter Analogread Serial
```

Starter Analog Input



Sensores

https://www.arduino.cc/reference/en/

- Botões*
- Potenciômetros* - Starter Arduino Analog read
- TMP36
- Sensor PIR - erbase18_SensorPir_Motor
- Sensor de distância por Ultrassom

TMP36

http://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf

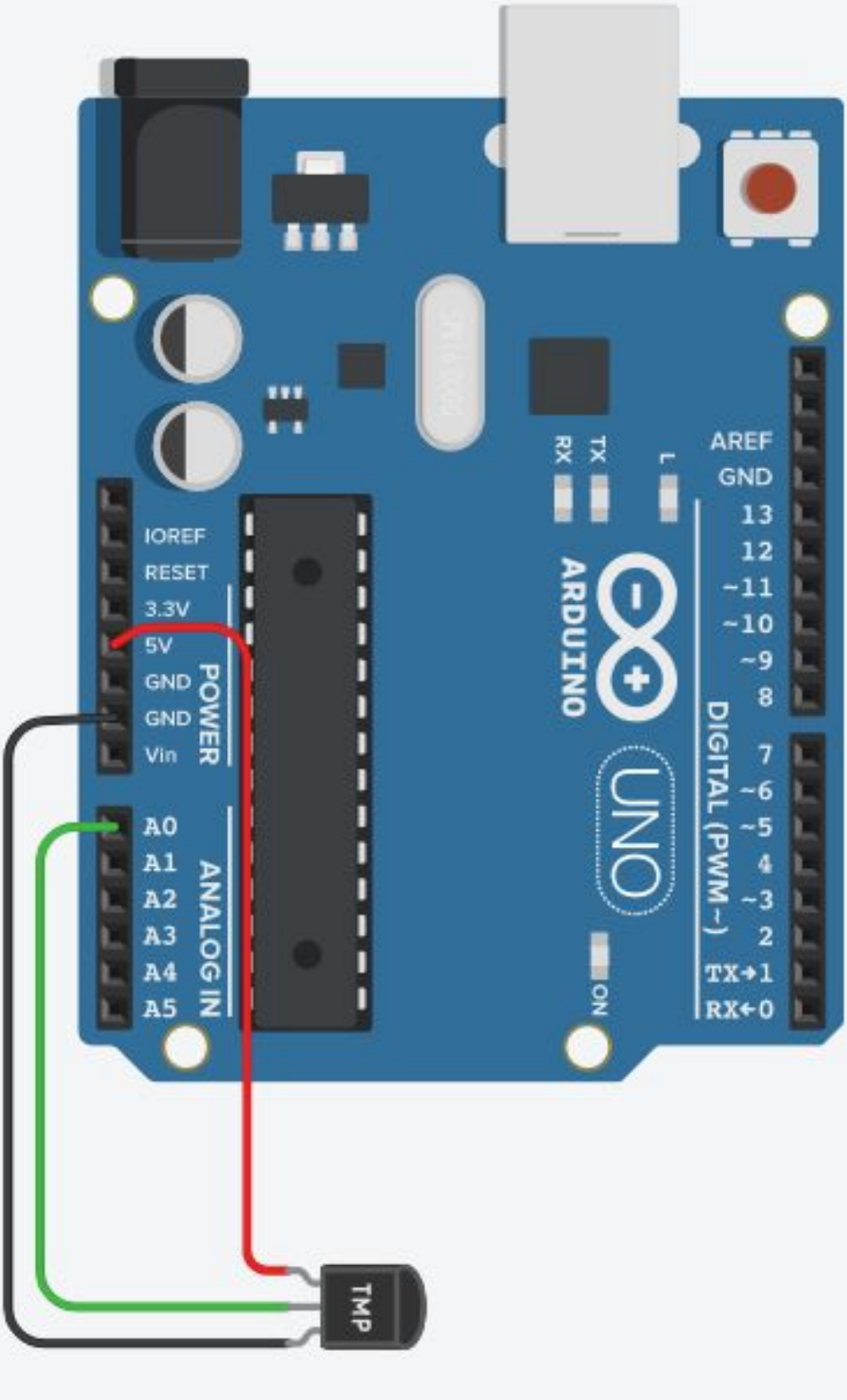
TMP36

10 mV/°C

−40°C a +125°C

```
int sensorValue = 0;

void setup(){
  pinMode(A0, INPUT);
  Serial.begin(9600);
}

void loop(){
  sensorValue = analogRead(A0);
  Serial.print(sensorValue);
  delay(10);
}
```

TMP36


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

TMP36

10 mV/°C

−40°C a +125°C

```
int sensorValue = 0;
int tempC = 0;

void setup(){
  pinMode(A0, INPUT);
  Serial.begin(9600);
}

void loop(){
  sensorValue = analogRead(A0);
  Serial.print(sensorValue);
  tempC = map(sensorValue,102,358,0,125);
  Serial.print(" - ");
  Serial.println(tempC);
  delay(10);
}
```

Sensores

https://www.arduino.cc/reference/en/

- Botões*
- Potenciômetros* - Starter Arduino Analog read
- TMP36
- Sensor PIR - erbase18_SensorPir_Motor
- Sensor de distância por Ultrassom

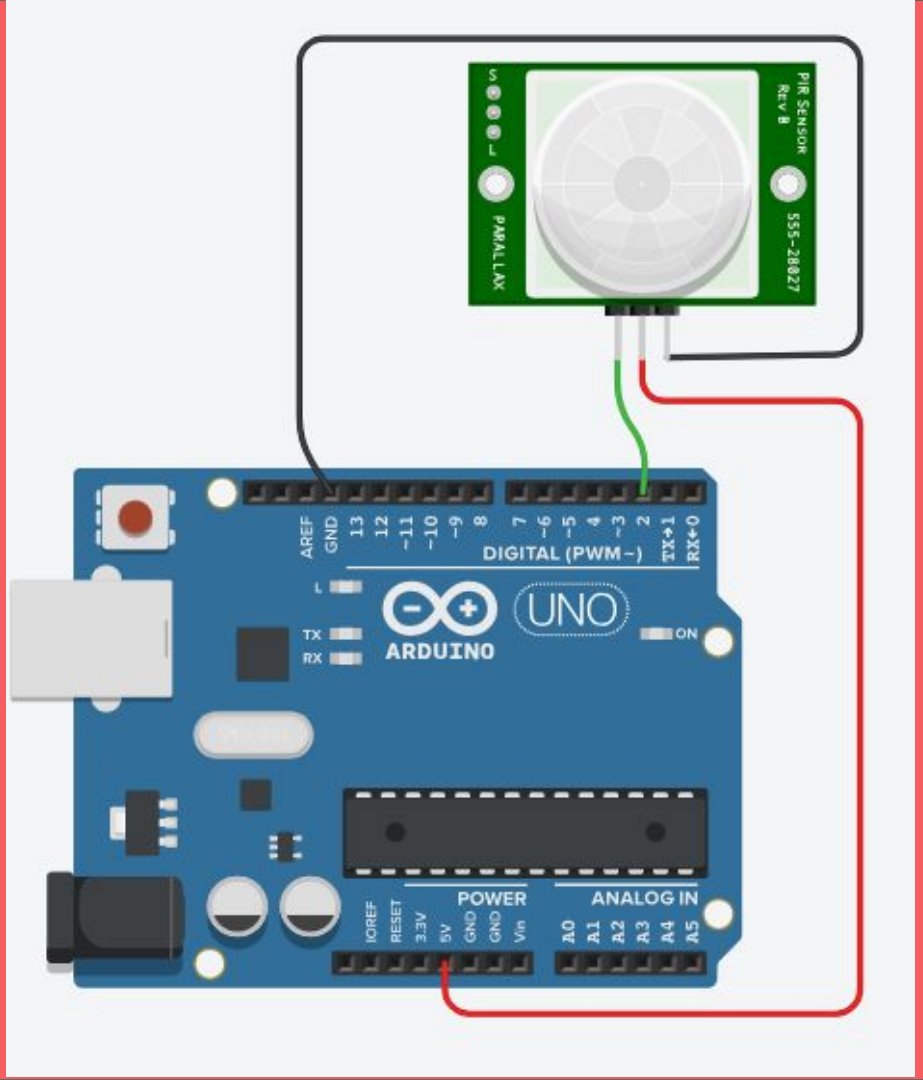
Sensor PIR

https://www.parallax.com/sites/default/files/downloads/555-28027-PIR-Sensor-Product-Guide-v2.3.pdf

http://learn.parallax.com/KickStart

PIR Sensor

3 a 6 V



PIR Sensor

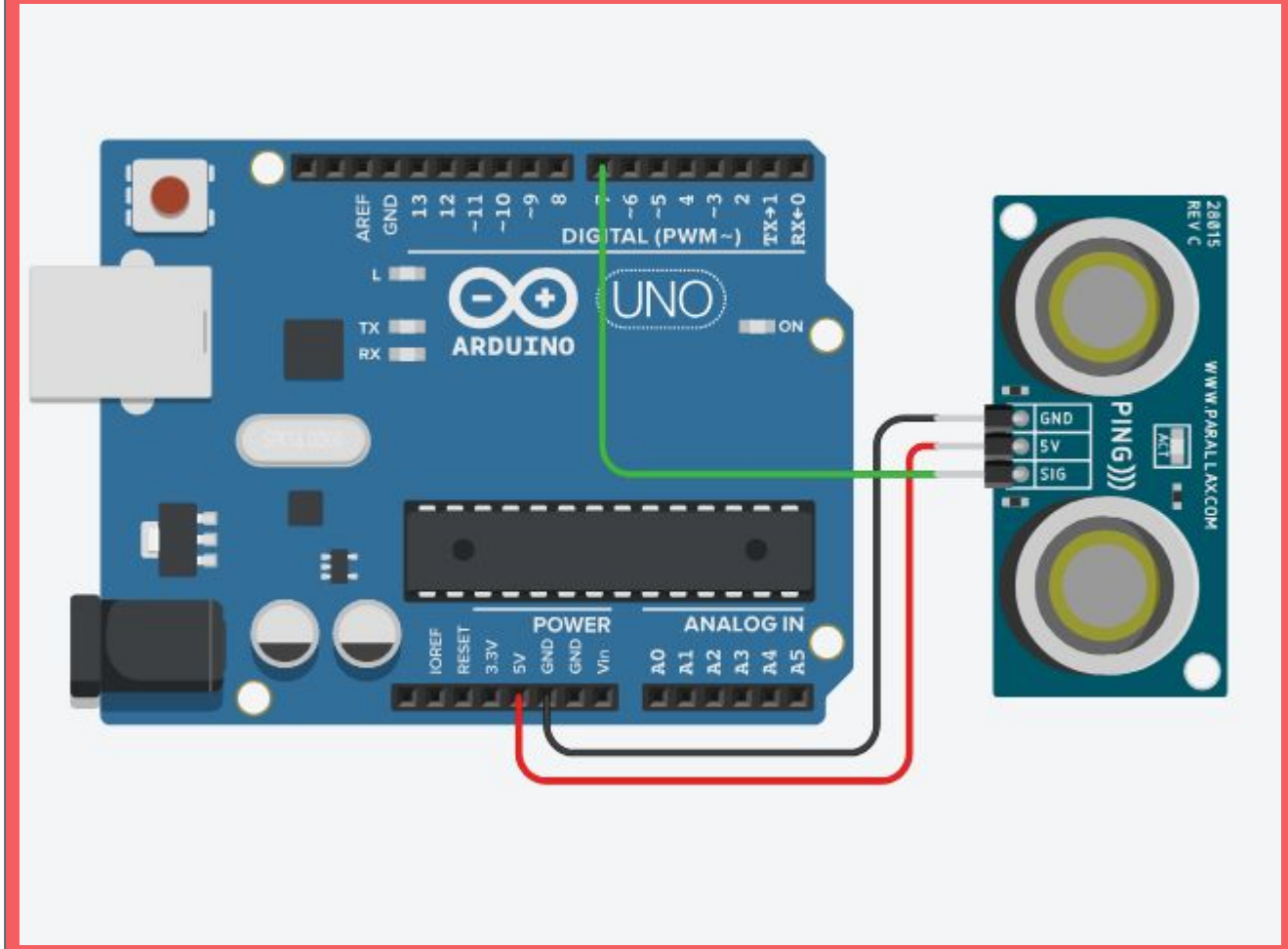
3 a 6 V

```
int sensorValue = 0;

void setup(){
  pinMode(2, INPUT);
  Serial.begin(9600);
}

void loop(){
  sensorValue = digitalRead(2);
  Serial.println(sensorValue);
  delay(10);
}
```

Sensor PING

https://www.parallax.com/product/28015
http://learn.parallax.com/KickStart

Sensor PING


AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
28015
REV C
WWW.PARALLAX.COM
PING)))
GND
5V
SIG

Sensor PING

3 a 6 V

```
const int ping = 7;
unsigned int duracao, cm;
void setup() {
Serial.begin(9600);
}
void loop() {
pinMode(ping, OUTPUT);
digitalWrite(ping, LOW);
delayMicroseconds(2);
digitalWrite(ping, HIGH);
delayMicroseconds(5);
digitalWrite(ping, LOW);
pinMode(ping, INPUT);
duracao = pulseIn(ping, HIGH);
cm = duracao * 0.01723;
Serial.println(cm);
delay(200);
}
```

Displays

https://www.arduino.cc/en/Reference/LiquidCrystal

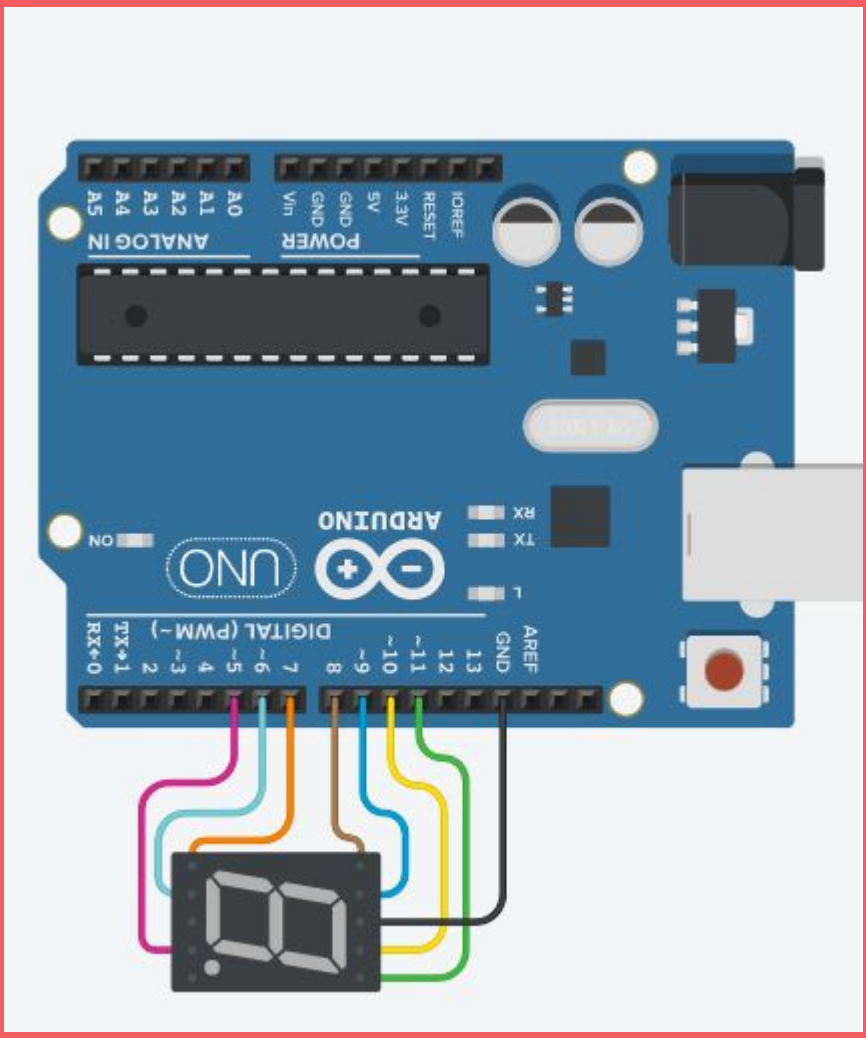
https://github.com/sigvaldm/SevenSeg/blob/master/extras/SevenSeg.pdf

- 7 Segmentos - erbase18_Display_7Segmentos
- LCD 16x2

Display de 7 segmentos

https://github.com/sigvaldm/SevenSeg
https://github.com/sigvaldm/SevenSeg/blob/master/extras/SevenSeg.pdf
https://www.tinkercad.com/things/dO6ZWkBRBZ9

Display de 7 segmentos



Display de 7 segmentos

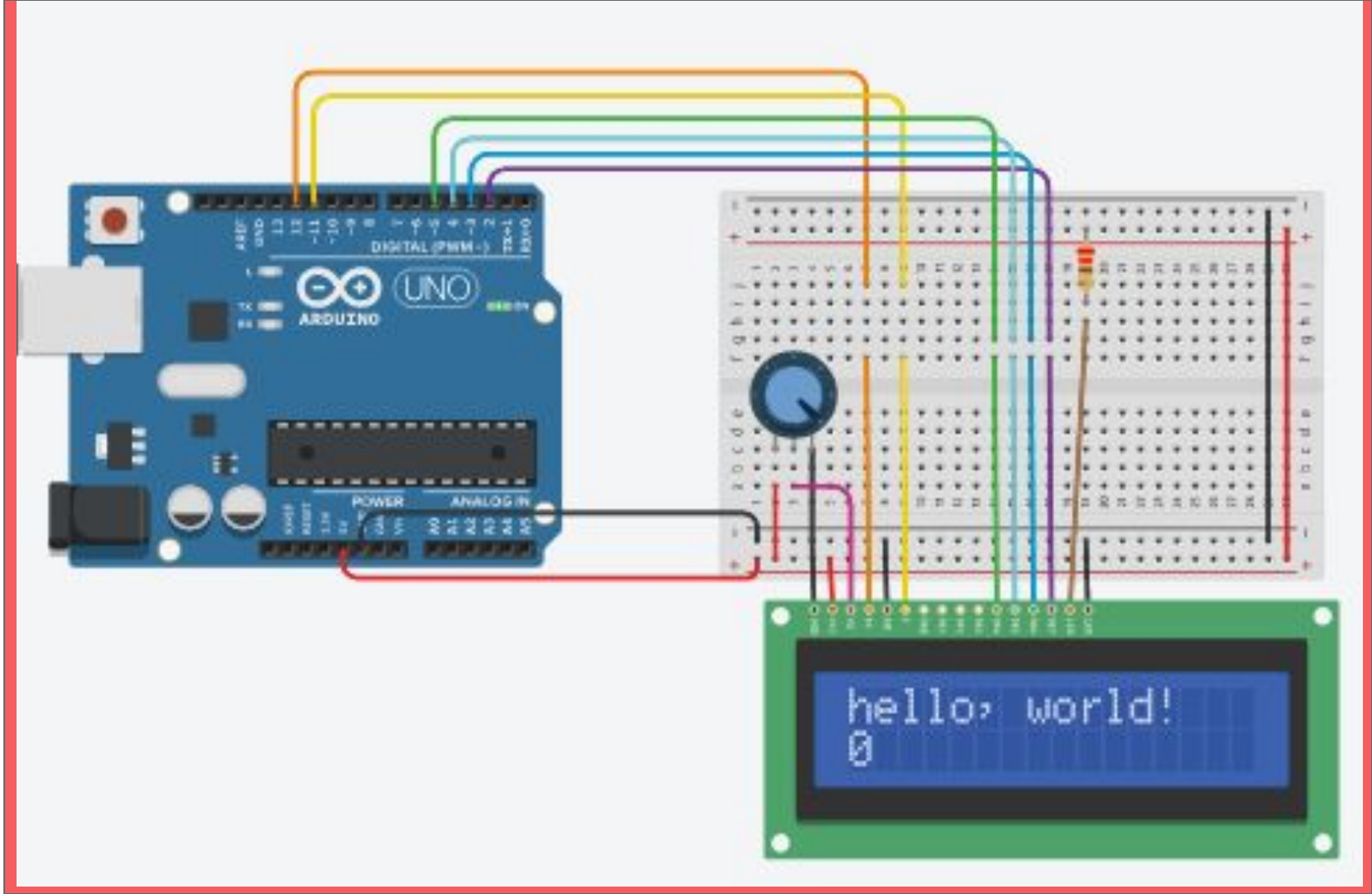
```
void display(int digit)
{
 //Digito 1
  if(digit == 1){

digitalWrite(b,HIGH);          //   --x--
digitalWrite(c,HIGH);          // |      |
                               // x      b
                               // |      |
                               //   --x--
                               // |      |
                               // x      c
                               // |      |
                               //   --x--
}
```

LCD

https://www.arduino.cc/en/Reference/LiquidCrystal

https://www.arduino.cc/en/Tutorial/HelloWorld

LCD



LCD

```
#include <LiquidCrystal.h> // biblioteca
// em quais pinos o lcd está ligado
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // tamanho do display linhas x colunas:
  lcd.begin(16, 2);
  // inicia na linha 0 coluna 0.
  // lcd.setCursor(0, 0);
  lcd.print("hello, world!");
}
```

Projeto Termômetro

Usando as ferramentas e componentes disponíveis no tinkercad desenvolva um termômetro que exibe a temperatura em um display.

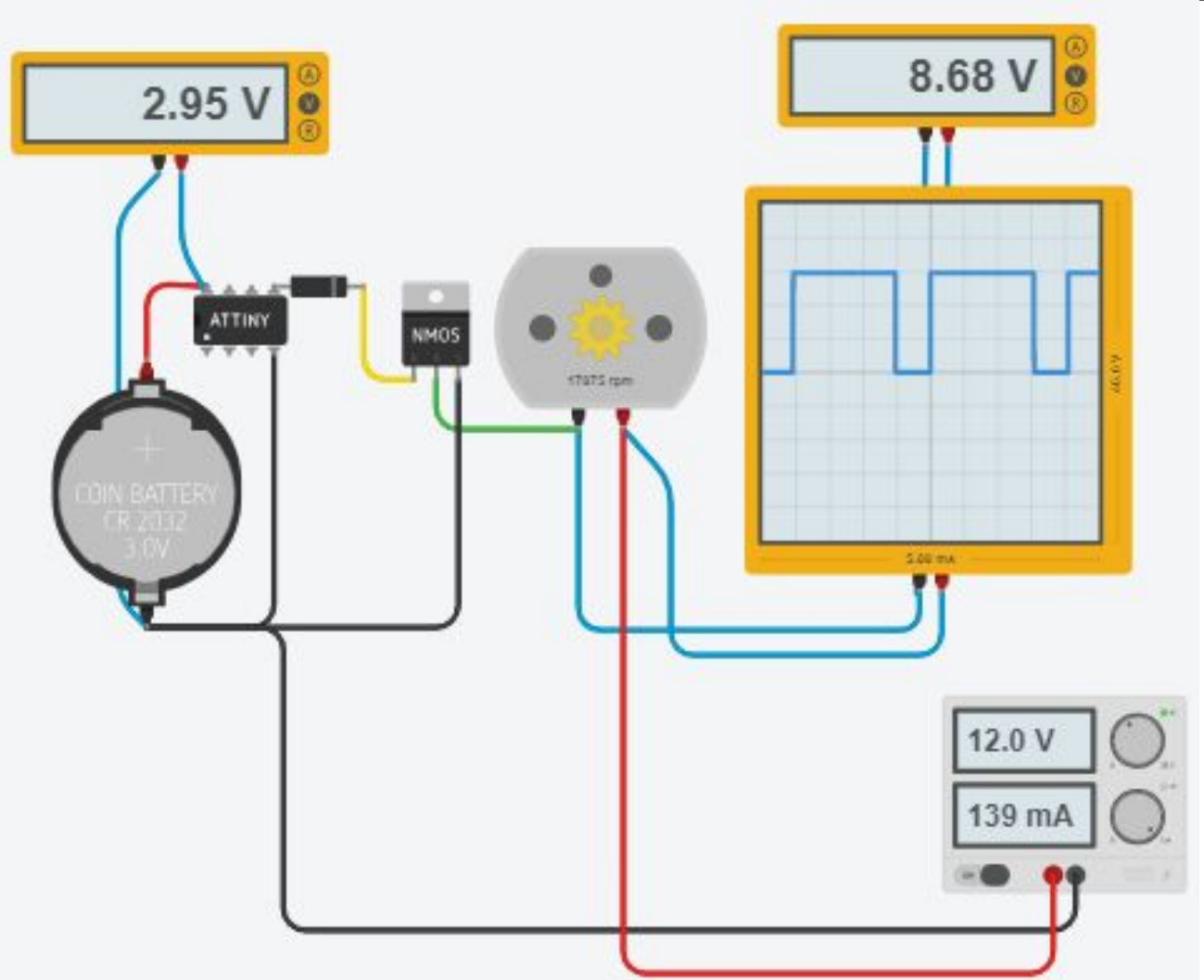
Motores, Servos e Solenoides

Atuadores mecânicos, movimentos e aplicações.

Motores

https://www.arduino.cc/en/Tutorial/TransistorMotorControl
https://www.tinkercad.com/things/5xMQqPYEgah
http://labdegaragem.com/profiles/blogs/tutorial-acionamento-de-motor-dc-com-transistor-tip122

Motor DC



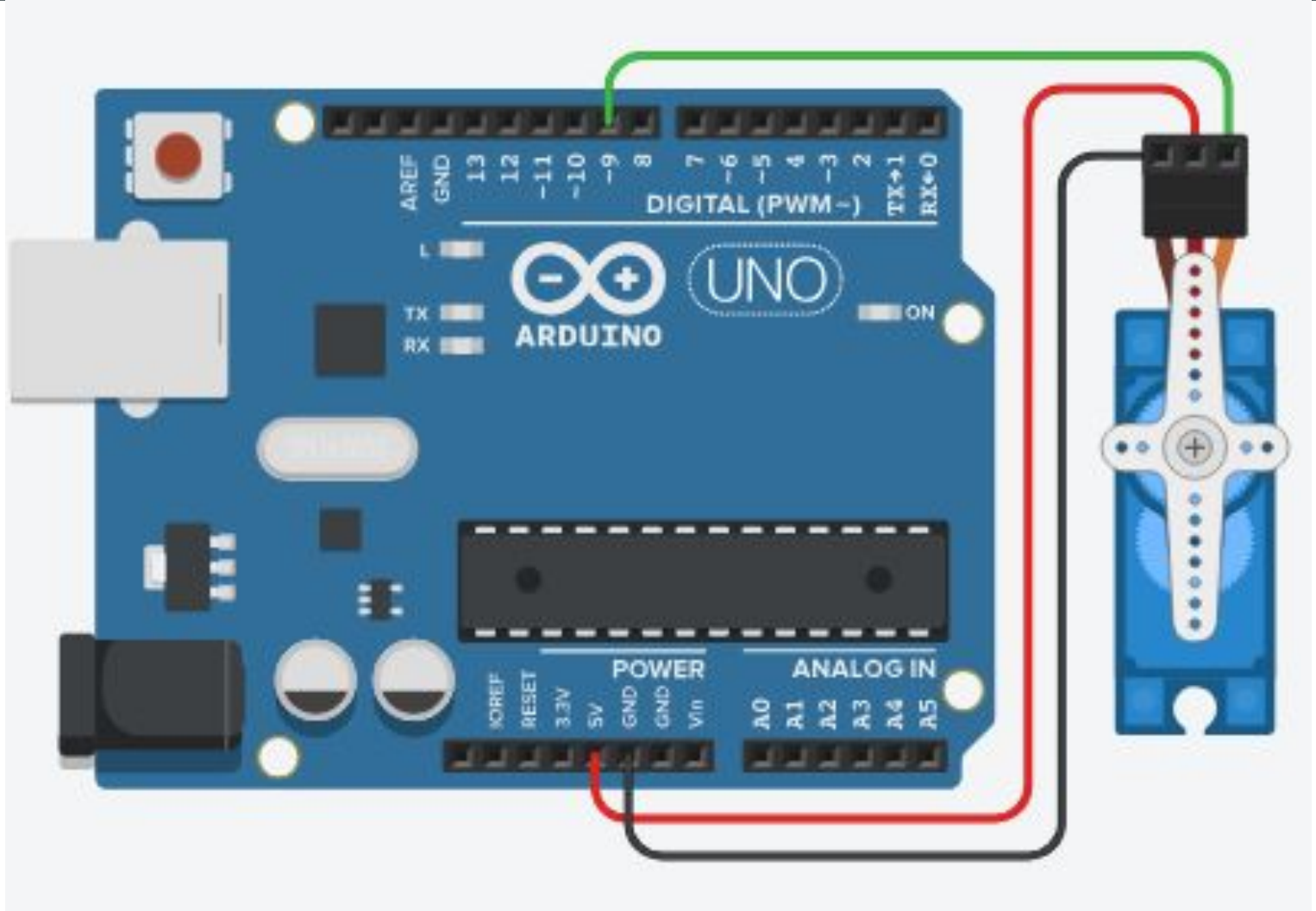
Motor DC


count up by 1 for i from 1 to 255 do
set pin 0 to i
wait 10 milliseconds
wait 10 secs

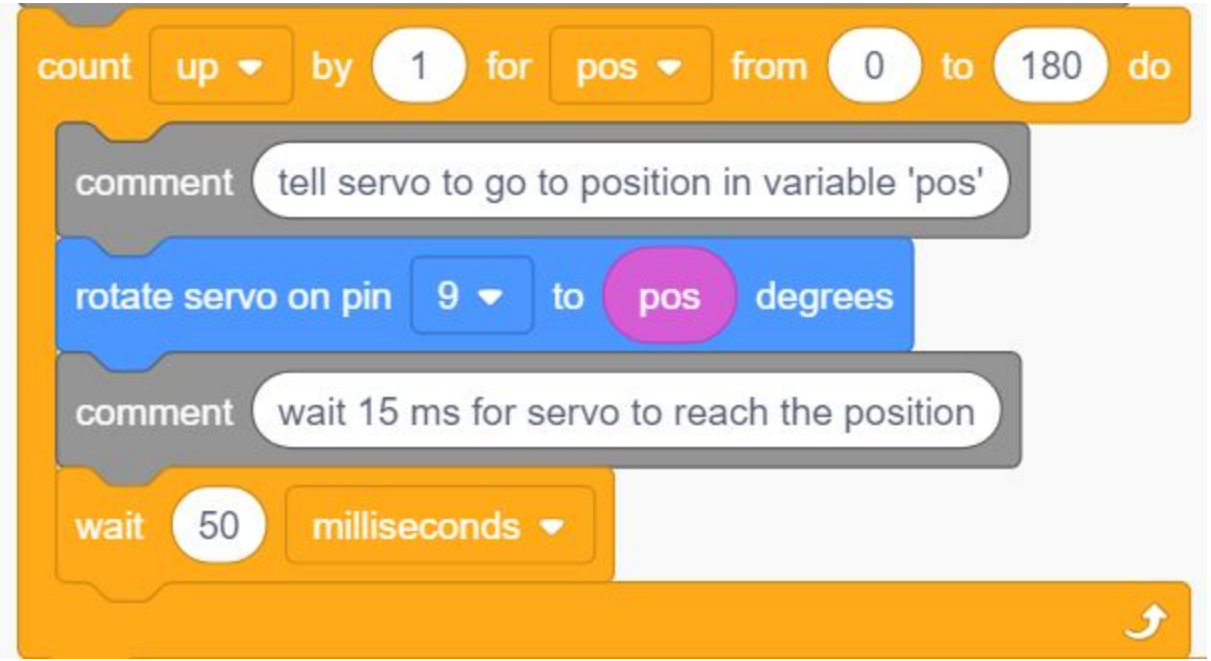
Servos

https://www.arduino.cc/en/Reference/Servo
https://www.arduino.cc/en/Tutorial/Knob

Servos


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

Servos

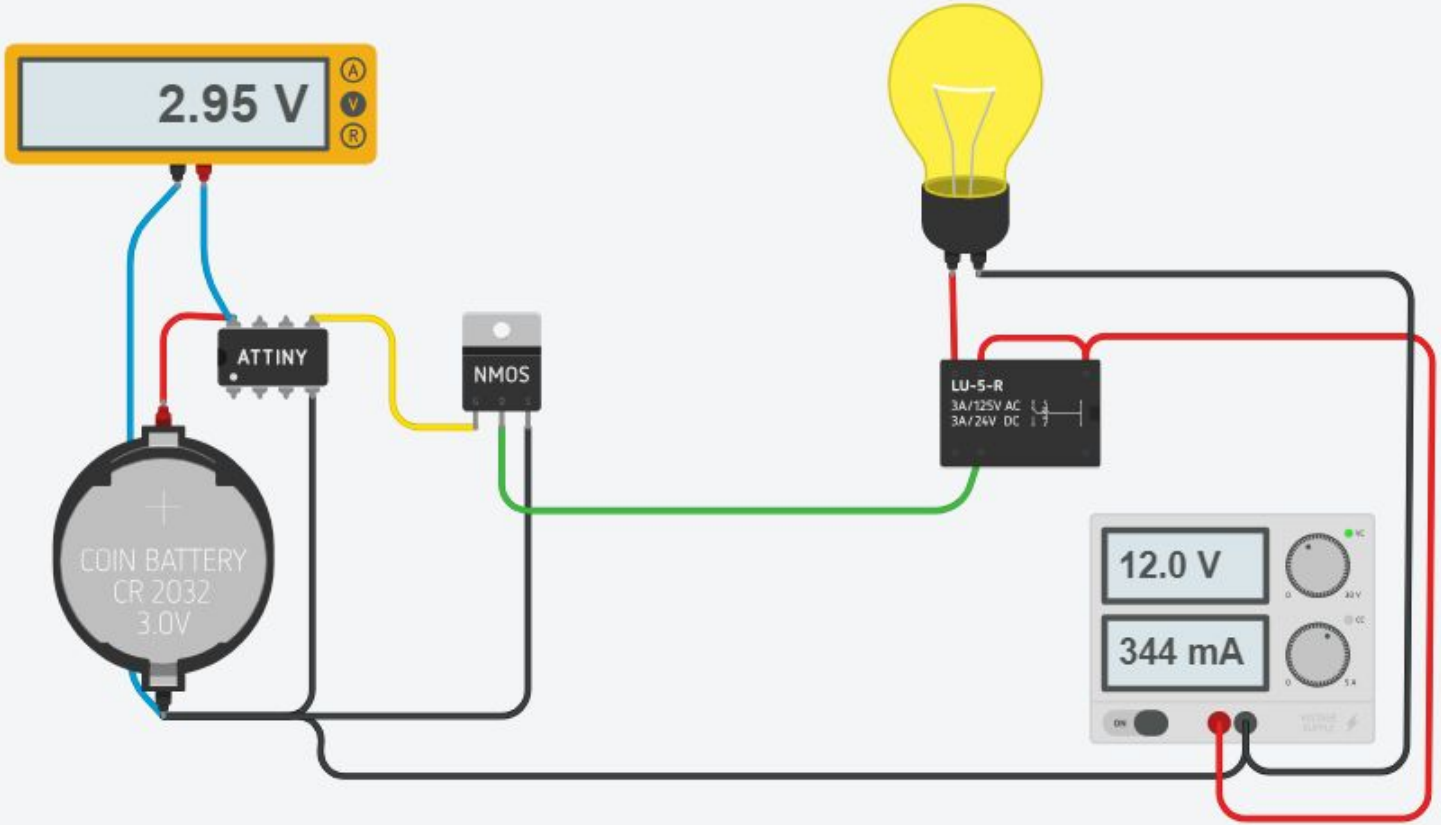

count up by 1 for pos from 0 to 180 do
comment tell servo to go to position in variable 'pos'
rotate servo on pin 9 to pos degrees
comment wait 15 ms for servo to reach the position
wait 50 milliseconds

Solenóides

https://playground.arduino.cc/Learning/SolenoidTutorial

o tinkercad não simula solenoides, mas este componente pode ser tratado como um motor, já que é baseado no mesmo princípio, transforma eletromagnetismo em movimento.

Solenóides

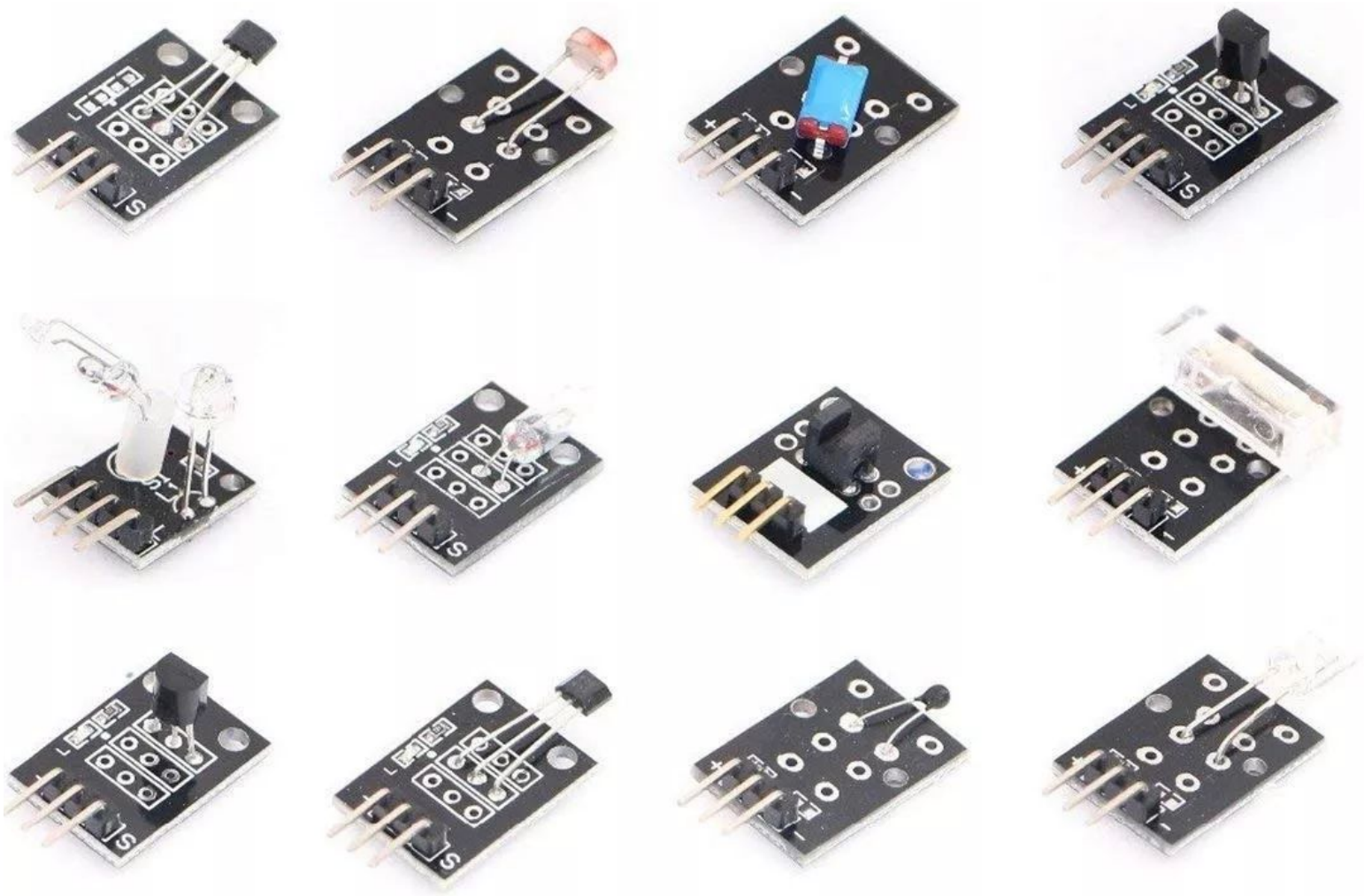

2.95 V
A
V
R
ATTINY
NMOS
LU-5-R
3A/125V AC
3A/24V DC
COIN BATTERY
CR 2032
3.0V
12.0 V
344 mA
ON

Expandindo a Capacidade do Arduino

Módulos, Shields e Bibliotecas

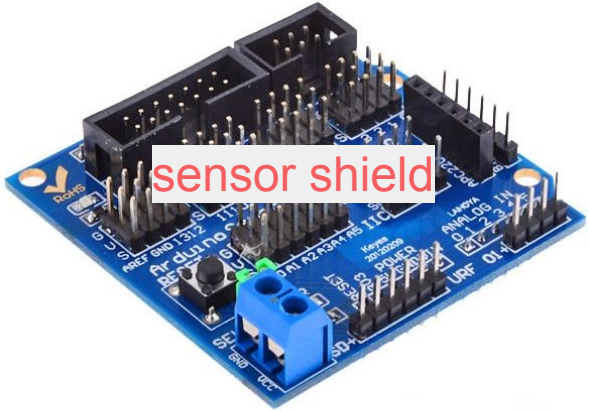
Módulos e Shields

módulos

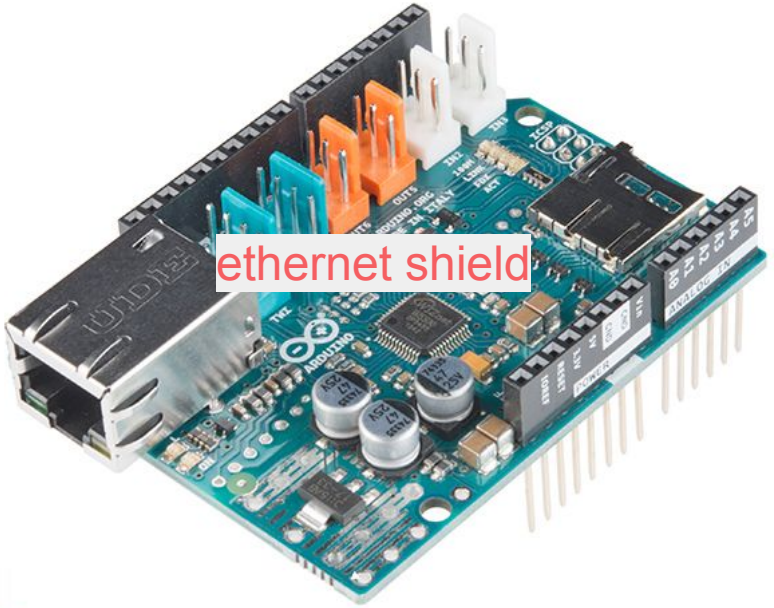


Shields

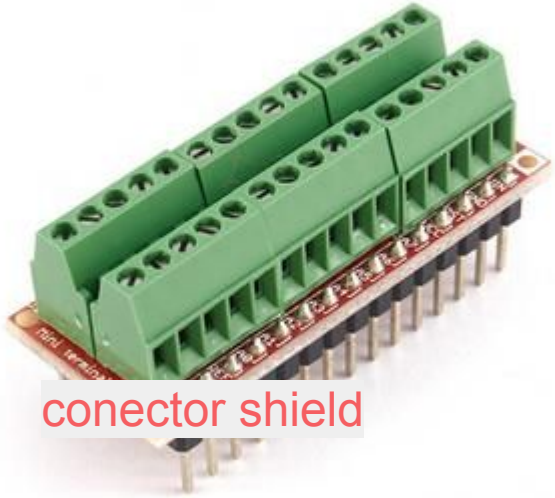
**http://shieldlist.org/



sensor shield



ethernet shield



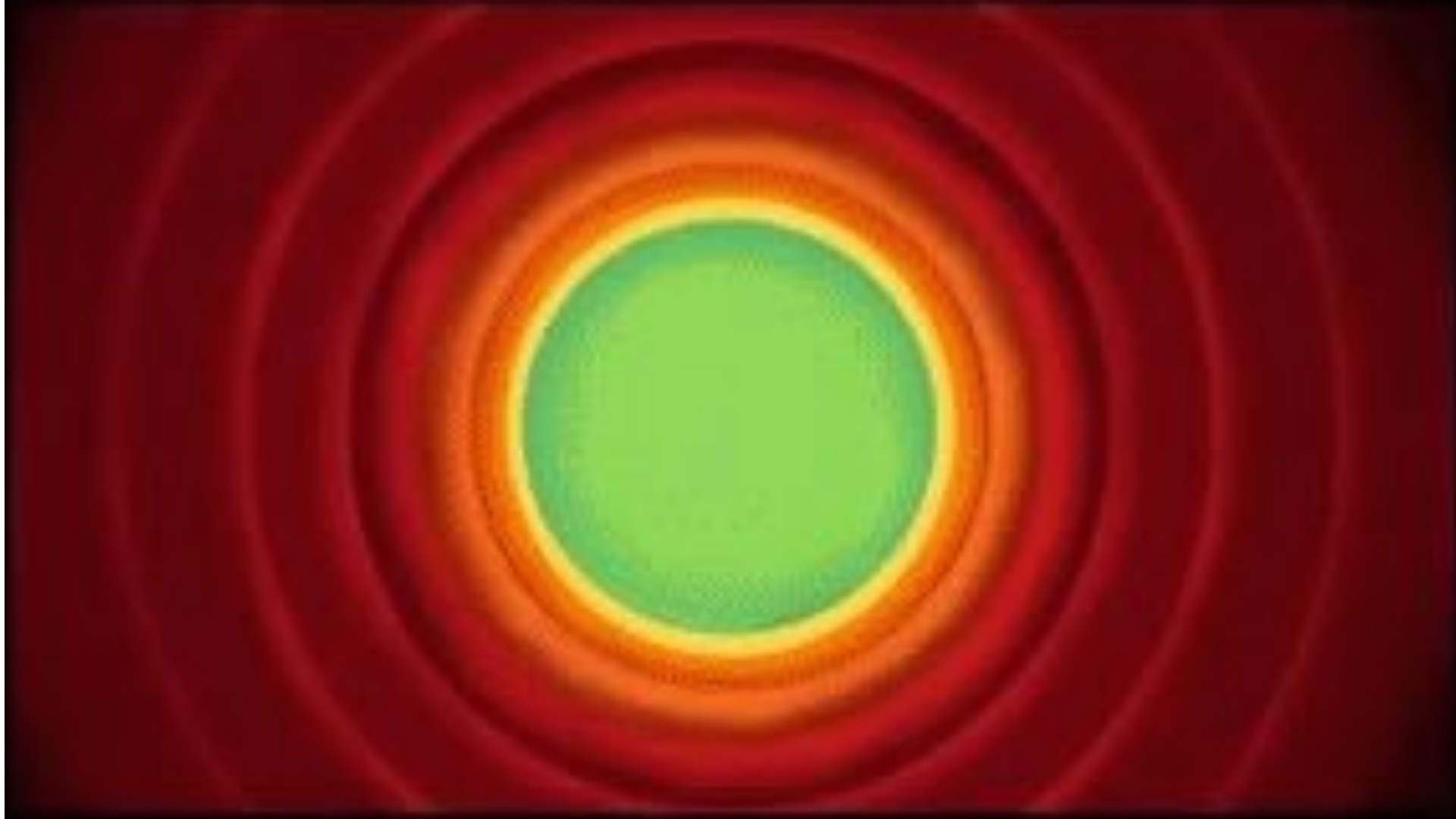
conector shield

Bibliotecas

https://www.arduino.cc/en/Reference/Libraries

Bibliotecas

EEPROM - leitura e gravação para armazenamento "permanente"
Ethernet - conexão à Internet usando o Arduino Ethernet Shield.
GSM - para conexão a uma rede GSM / GRPS com o Shield GSM.
LiquidCrystal - displays de cristal líquido (LCDs)
SD - para ler e escrever cartões SD
Servo - para controlar servomotores
SoftwareSerial - para comunicação serial em qualquer pino digital.




erbase
2018