

Instrumentação e Aquisição de Dados

LEFT, 2023-2024

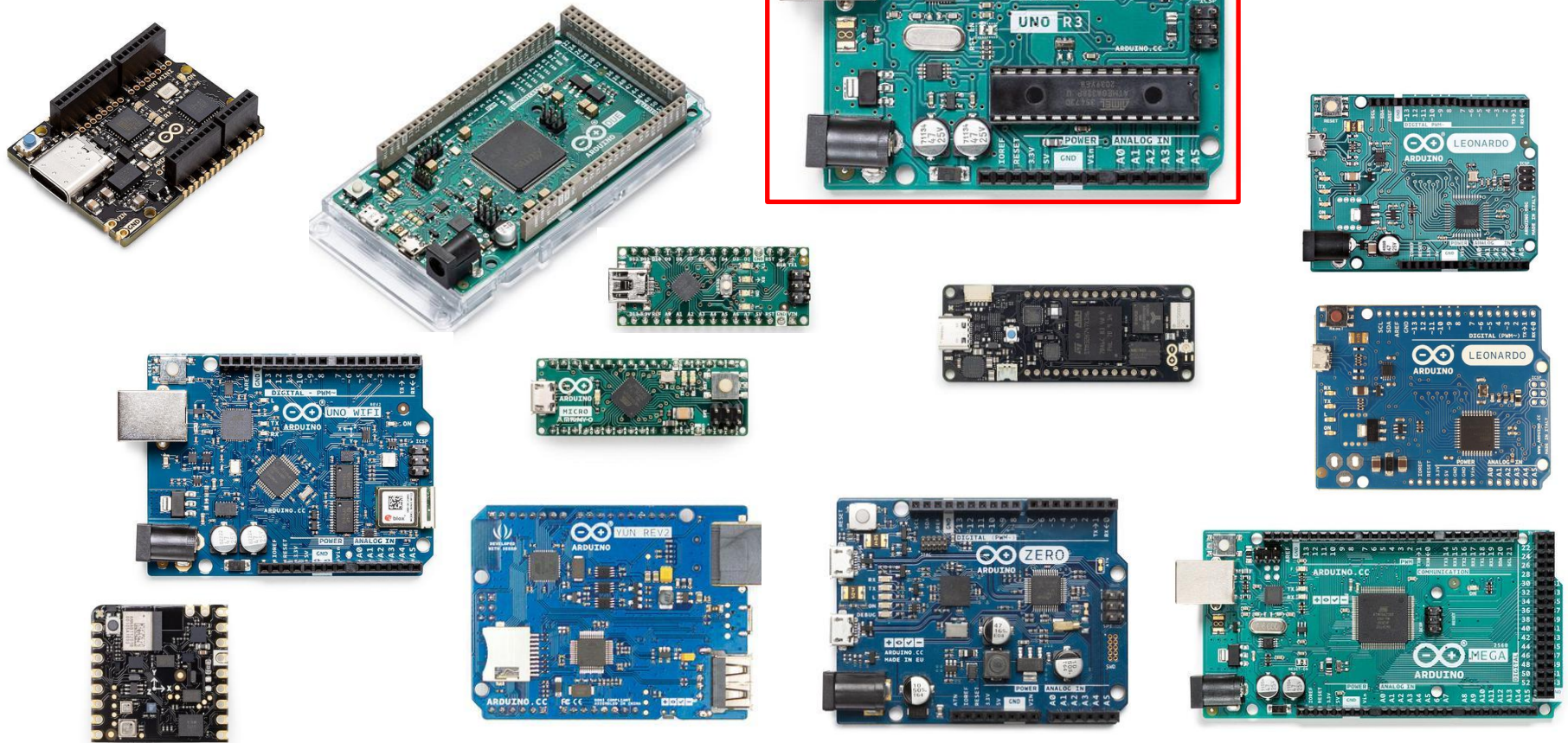
Arduino

- O Arduino é uma plataforma electrónica aberta baseada em hardware e software fáceis de utilizar (www.arduino.cc).
- Um Arduino é capaz de ler inputs (sensores, botões...) e activar outputs (ligar um led, ligar um motor, enviar mensagens – por exemplo por USB)
- A programação é feita utilizando uma linguagem própria (subset de C, sobre Wiring).
- Utiliza-se um ambiente de programação próprio (Arduino IDE) num PC (Windows), MAC (MAC OS), ou linux, que compila o programa (sketch) e carrega no Arduino por USB
- A partir daí, o programa corre no Arduino standalone



Arduino

- Há diferentes modelos, com diferentes características, desde o número de entradas/saídas, processador, dimensões, wifi, custo, etc.
- Vamos usar o Arduino UNO



Arduino

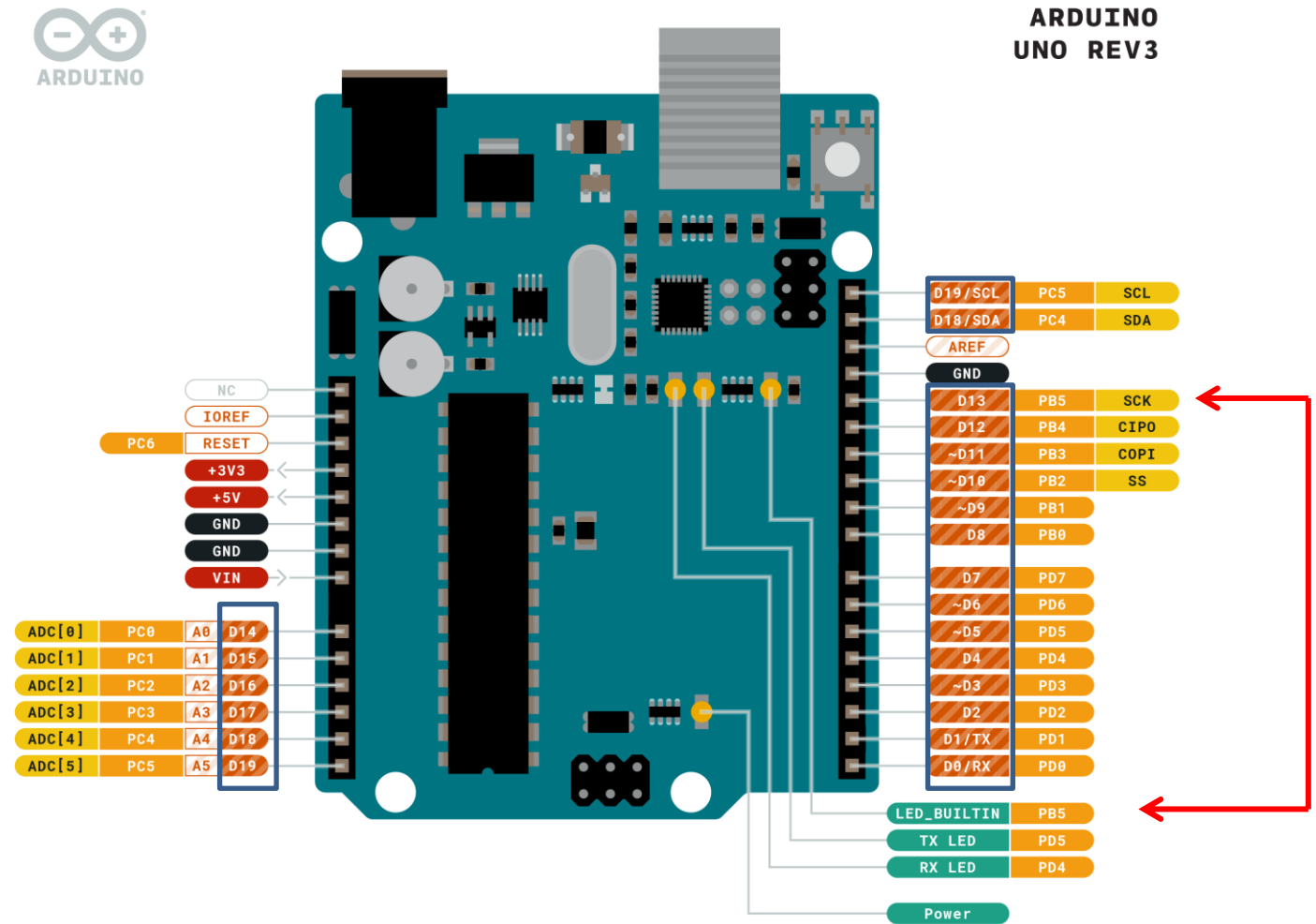
- Pinos de entrada/saída: 14 (configuráveis)
- 6 podem ser configurados para PWM (Pulse-Width Modulation)
- 6 podem ser configurados como ADCs (10 bits)
- 2 podem ser configurados para I2C
- 4 podem ser configurados para SPI



Arduino



- Pinout:



Ground	Internal Pin	Digital Pin	Microcontroller's Port
Power	SWD Pin	Analog Pin	
LED	Other Pin	Default	

ARDUINO . CC

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Arduino

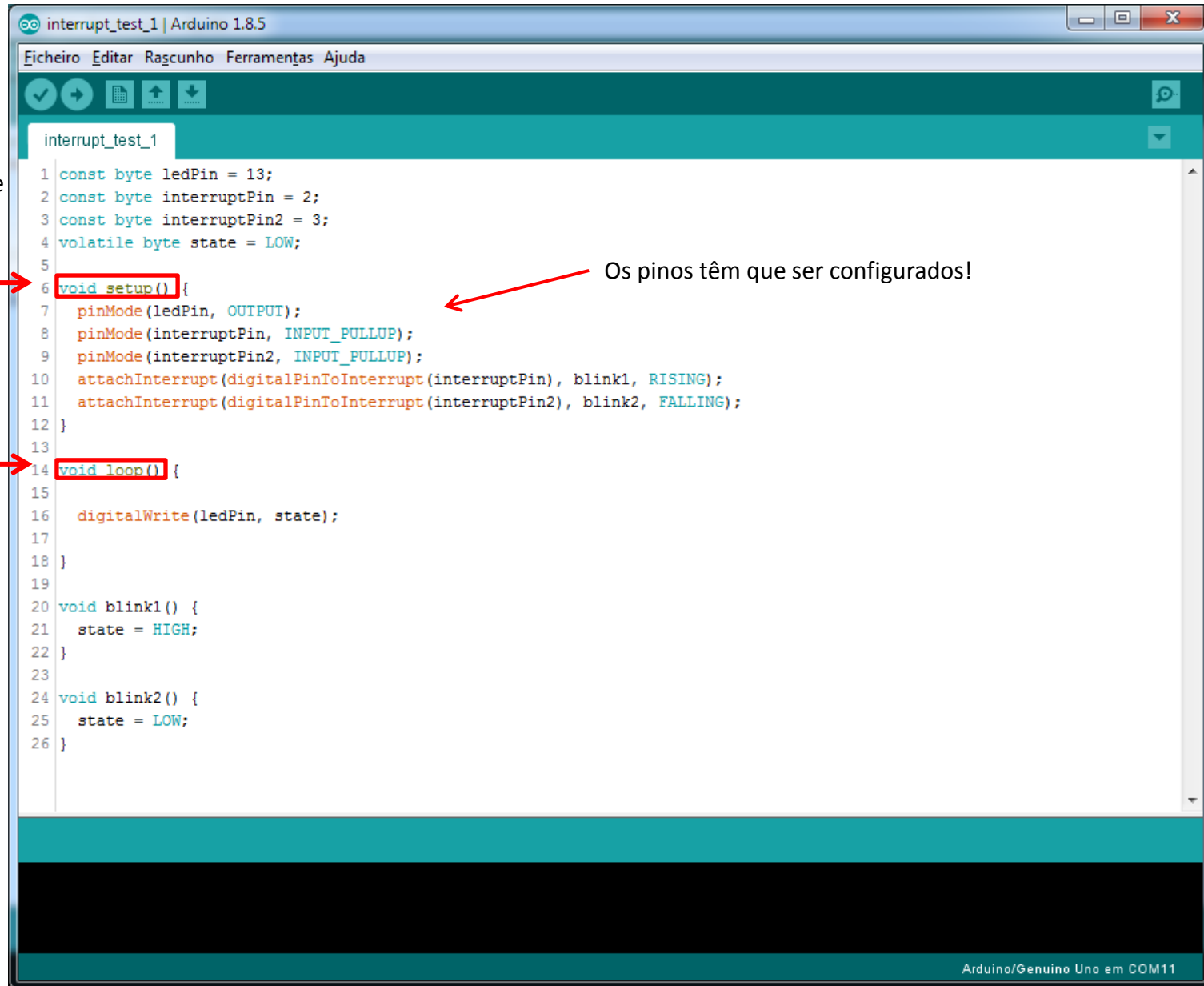
- Arduino IDE1:
<https://www.arduino.cc/en/software>

Obrigatória:

Corre no
arranque

Obrigatória:

Loop de
execução



```
interrupt_test_1 | Arduino 1.8.5
Ficheiro Editar Rasgunho Ferramentas Ajuda

interrupt_test_1
1 const byte ledPin = 13;
2 const byte interruptPin = 2;
3 const byte interruptPin2 = 3;
4 volatile byte state = LOW;
5
6 void setup() {
7   pinMode(ledPin, OUTPUT);
8   pinMode(interruptPin, INPUT_PULLUP);
9   pinMode(interruptPin2, INPUT_PULLUP);
10  attachInterrupt(digitalPinToInterrupt(interruptPin), blink1, RISING);
11  attachInterrupt(digitalPinToInterrupt(interruptPin2), blink2, FALLING);
12 }
13
14 void loop() {
15
16   digitalWrite(ledPin, state);
17
18 }
19
20 void blink1() {
21   state = HIGH;
22 }
23
24 void blink2() {
25   state = LOW;
26 }
```

Os pinos têm que ser configurados!

Arduino/Genuino Uno em COM11

Arduino

- Linguagem : (<https://docs.arduino.cc/hardware/uno-rev3> -> Essentials -> Language Reference)

The screenshot displays the Arduino Reference website in a web browser. The browser's address bar shows the URL `arduino.cc/reference/en/?_gl=1*t7p8g6*_ga*MTM3MDQ1ODA4MS4xNjAzMjE2NDkz*_ga_NEXN8H46L...`. The website's navigation bar includes links for **HARDWARE**, **SOFTWARE**, **CLOUD**, **DOCUMENTATION**, **COMMUNITY**, **BLOG**, and **ABOUT**. On the left sidebar, a menu lists **LANGUAGE** (with sub-items: FUNCTIONS, VARIABLES, STRUCTURE), **LIBRARIES**, **IOT CLOUD API**, and **GLOSSARY**. Below this menu, there is a notice about the Creative Commons Attribution-Share Alike 3.0 License and a link to suggest improvements on GitHub. A promotional image for 'The Arduino® Student Kit: bring the buzz home' is also visible. The main content area is titled **Functions** and describes their purpose: 'For controlling the Arduino board and performing computations.' It lists various function categories and their respective functions:

- Digital I/O**: `digitalRead()`, `digitalWrite()`, `pinMode()`
- Analog I/O**: `analogRead()`, `analogReference()`, `analogWrite()`
- Zero, Due & MKR Family**: `analogReadResolution()`, `analogWriteResolution()`
- Advanced I/O**: `noTone()`, `pulseIn()`, `pulseInLong()`, `shiftIn()`, `shiftOut()`, `tone()`
- Time**: `delay()`, `delayMicroseconds()`, `micros()`, `millis()`
- Math**: `abs()`, `constrain()`, `map()`, `max()`, `min()`, `pow()`, `sq()`, `sqrt()`
- Trigonometry**: `cos()`, `sin()`, `tan()`
- Characters**: `isAlpha()`, `isAlphaNumeric()`, `isAscii()`, `isControl()`, `isDigit()`, `isGraph()`, `isHexadecimalDigit()`, `isLowerCase()`, `isPrintable()`, `isPunct()`, `isSpace()`, `isUpperCase()`, `isWhitespace()`
- Random Numbers**: `random()`, `randomSeed()`
- Bits and Bytes**: `bit()`, `bitClear()`, `bitRead()`, `bitSet()`, `bitWrite()`, `highByte()`, `lowByte()`
- External Interrupts**: `attachInterrupt()`, `detachInterrupt()`
- Interrupts**: `interrupts()`, `noInterrupts()`
- Communication**: `Serial`, `Stream`
- USB**: `Keyboard`, `Mouse`