

# ARDUINO

Níveis lógicos de referência  
VCC e GND

Entradas e saídas  
digitais e analógicas

# Positivo e negativo

Nos esquemas e dispositivos (módulos, arduinos) é comum usar os nomes:

- **VCC (+)**, ou **V+**

(Voltage = tensão;  
CC é sufixo usado  
por convenção)

- **GND (-)**

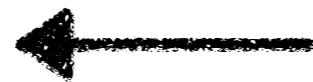
(Grou**ND** = terra =  
pólo negativo = 0  
volts)

Referência é **relativa!**

Tensão é **DIFERENÇA** de potencial



**VCC para GND (+)**



**GND para VCC (-)**

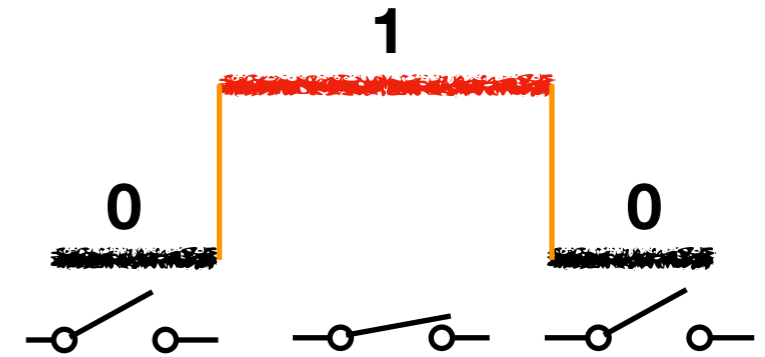


**GND para GND (0)**



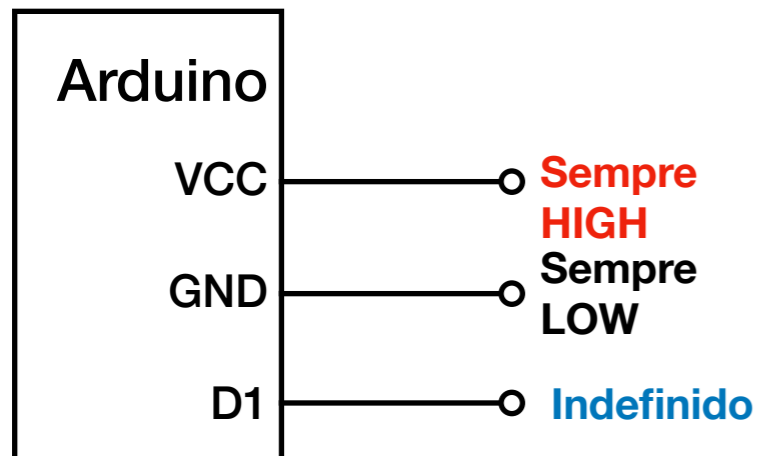
**VCC para VCC (0)**

# Lógica digital



- Associa **valores de tensão** a **níveis lógicos** binários (0 ou 1)
- O valor da tensão é uma **referência**. A referência mais comum para o valor **ALTO** (HIGH) ou **LIGADO** (ON) é **5V**
- A referência para o valor **BAIXO** (LOW) ou **DESLIGADO** (OFF) é **0V**
- Em programação, **5V** é representado pelo algarismo **1** e **0V** é representado pelo algarismo **0**
- **Informação** é representada em **programas de computador** através de **sequências de 0s e 1s** (lógica binária). A unidade é o **bit** (que pode ser 0 ou 1) e palavras (sequências) de 8 bits são **bytes**.
- **Bits** são representados em **eletrônica** por **chaves** (geralmente transistores) que podem reter 5V (armazenam a informação 1) ou 0V (armazenam 0). A lógica também pode ser inversa.

# Níveis lógicos (entrada)



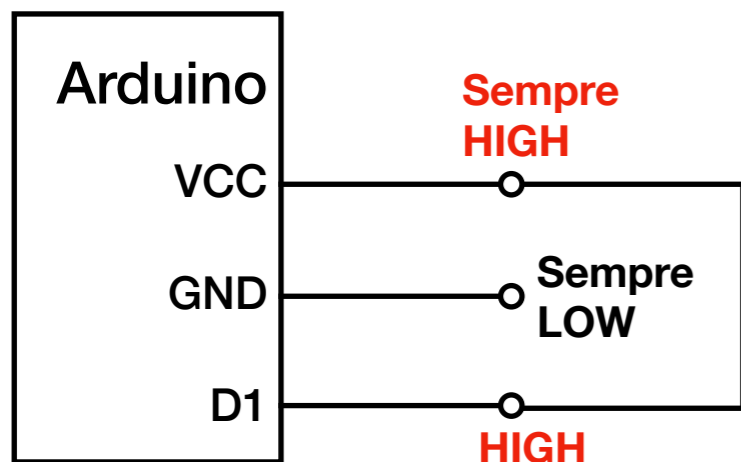
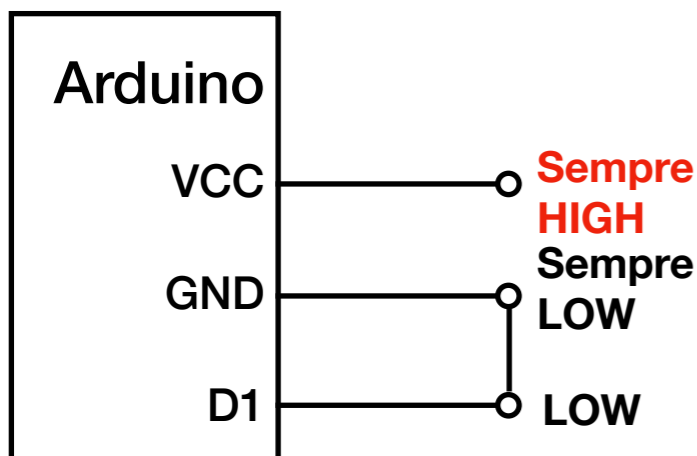
Pino **VCC** (Pino 5V em Arduino Uno/Nano) sempre fornece nível lógico **HIGH**

Pino **GND** sempre fornece nível lógico **LOW**

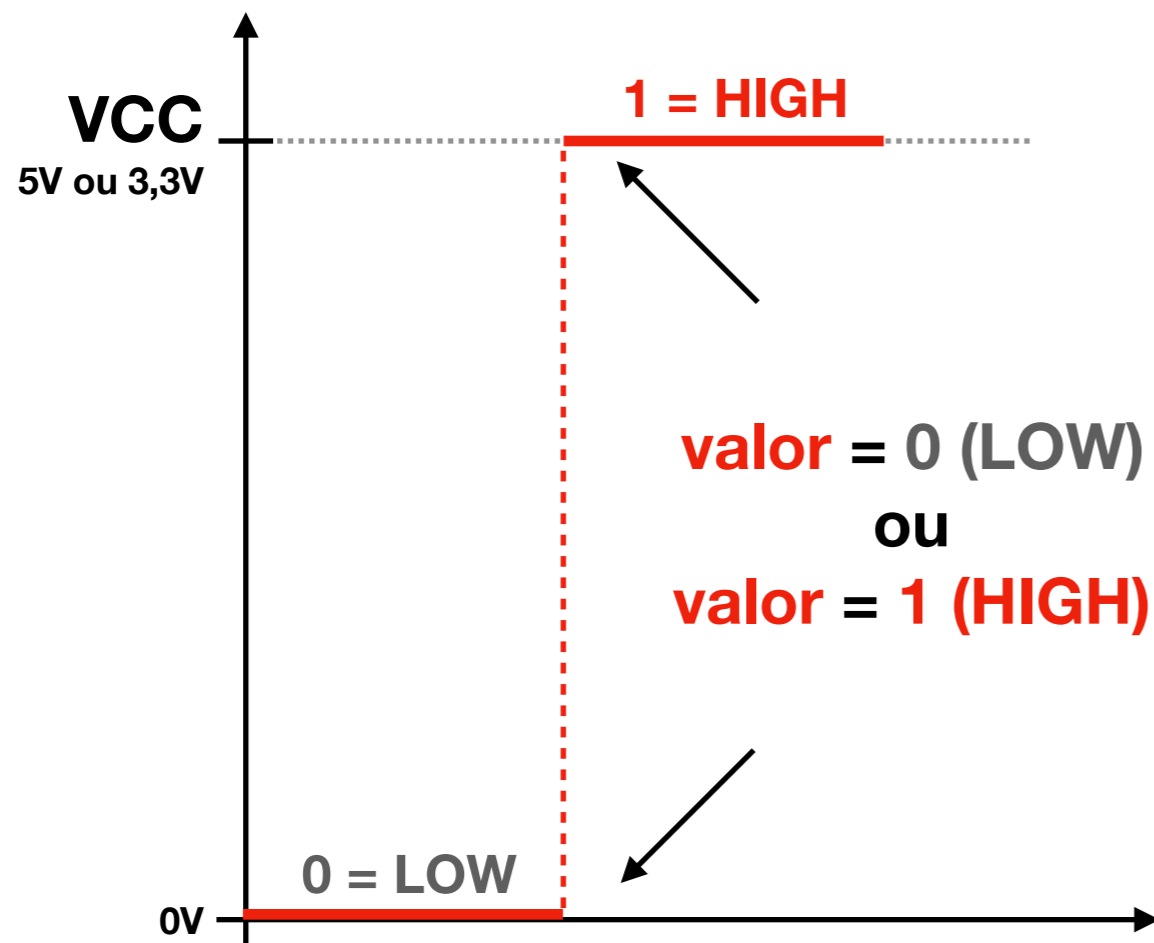
Pinos digitais **D1** a **D19** têm níveis lógicos indefinidos (podem ser interpretados como **HIGH** ou **LOW**)

Pino digital configurado como **INPUT** (default) pode ser ligado a **VCC (HIGH)** ou **GND (LOW)** para ter valor definido

Ligação pode ser direta (fio) ou via resistor de **pull-up** (para **VCC**) ou pull-down (para **GND**)



# int **valor** = digitalRead(pino)



**VCC** representa o nível lógico **HIGH** do Arduino (5V ou 3,3V dependendo do tipo)

Se **VCC** for 5V (Arduino Uno)

**LOW** = 0V, **HIGH** = 5V

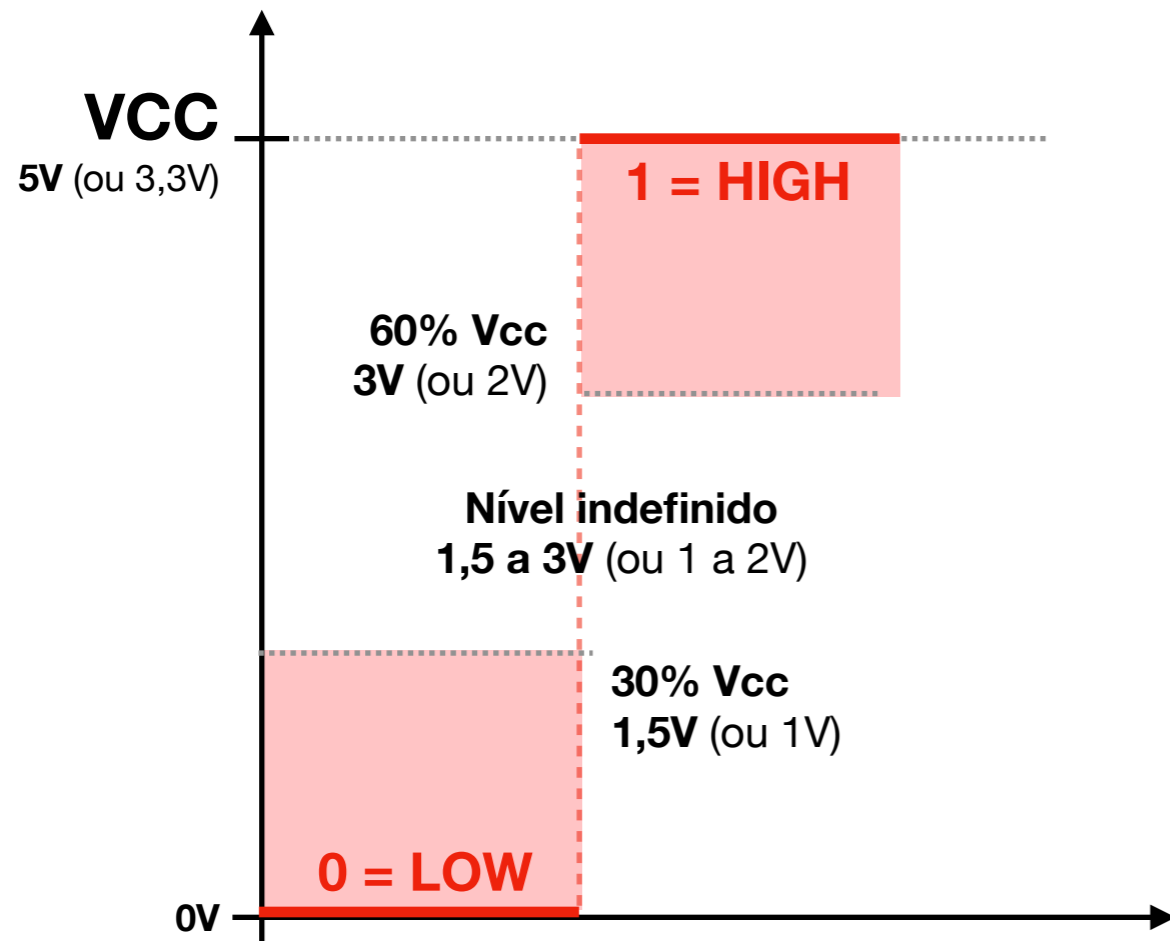
Se **VCC** for 3,3V (Due, LilyPad, ProMini)

**LOW** = 0V, **HIGH** = 3,3V

Não há como mudar o nível **VCC** do Arduino. Ele é fixo. Cada pino em nível **HIGH** terá esse valor.

Para conectar componentes com níveis lógicos incompatíveis (3,3V x 5V) é preciso usar um **conversor de nível lógico** (pode ser um divisor de tensão)

# Níveis lógicos reais (entrada)

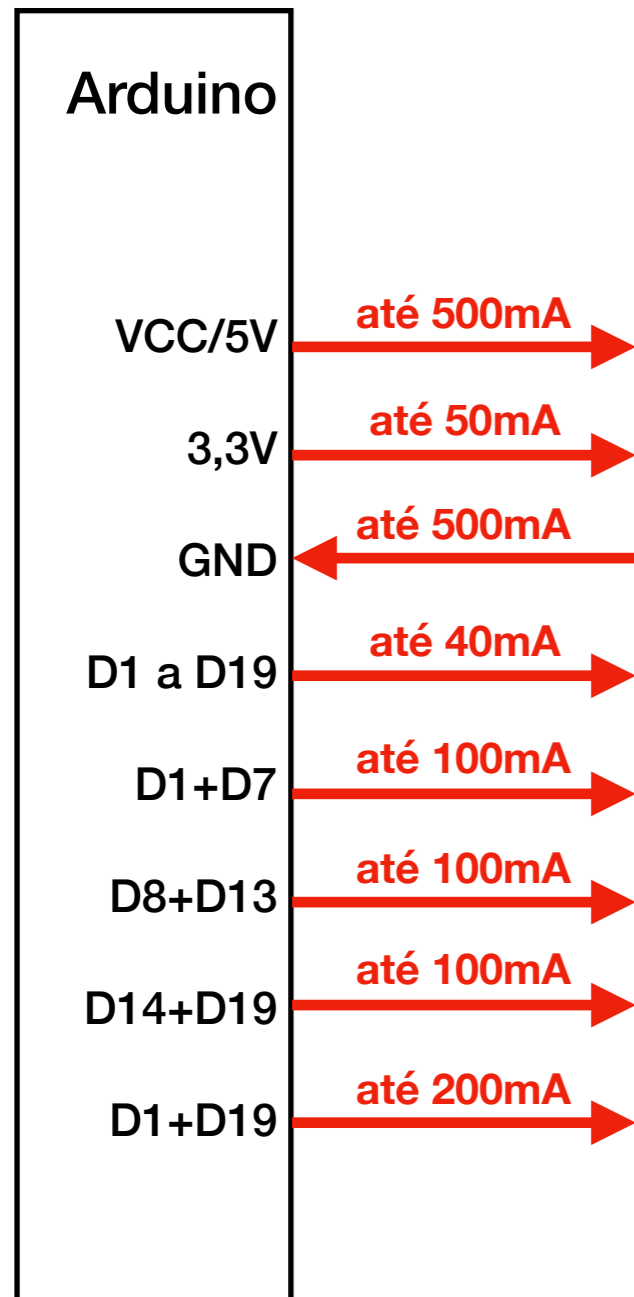


Um valor de tensão é traduzido para o nível lógico 1 (HIGH) se for maior que **60% da tensão VCC** do Arduino

Um valor de tensão é traduzido para o nível lógico 0 (LOW) se for menor que **30% da tensão VCC** do Arduino

Valores intermediários são **indefinidos** e serão traduzidos para **qualquer um** dos dois níveis lógicos (não há como prever)

# Níveis lógicos (saída)



Nível lógico em pino configurado com **pinMode OUTPUT** é traduzido para uma **tensão** de valor **VCC** (5V ou 3,3V) ou 0V.

Para **usar a tensão** produzida por um pino **OUTPUT** em um circuito é preciso observar os **limites de corrente\***:

~40mA para qualquer **pino** digital

~100mA para pinos de um **grupo** (D1-D7, D8-D13, A0-A5)

~200mA para o **Arduino inteiro** (vários pinos)

~50mA para o **pino 3,3V** (nos modelos de 5V)

~500mA para o pino **VCC/5V** ou **GND** (limite USB)

**ATENÇÃO:** Ligar **OUTPUT** com nível **LOW** diretamente a **VCC**, ou **OUTPUT** com nível **HIGH** diretamente a **GND** sem limitar a corrente produz um curto-circuito que **queima o pino**.

Use um resistor calculado para limitar a corrente demandada pelo circuito de saída quando a diferença de potencial entre o pino e **VCC** ou **GND** for máxima (5 volts)

\* valores **típicos** do Uno - valores são diferentes em outros modelos e clones

# digitalWrite(pino, **valor**)

Valor é nível lógico (**1 = HIGH**, **0 = LOW**)

Nível lógico é **traduzido** para uma **tensão**:

**HIGH**

**5V** em Arduino Uno, Nano, Mega

**3,3V** em Arduino Due, LilyPad\*, ProMini\*

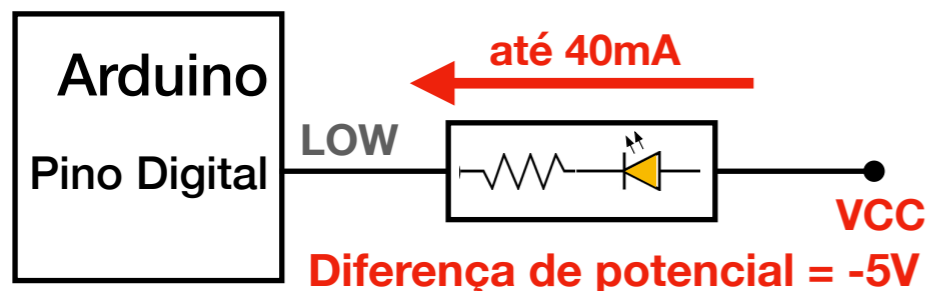
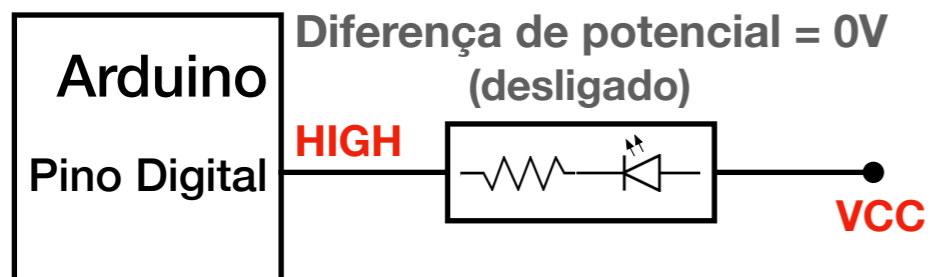
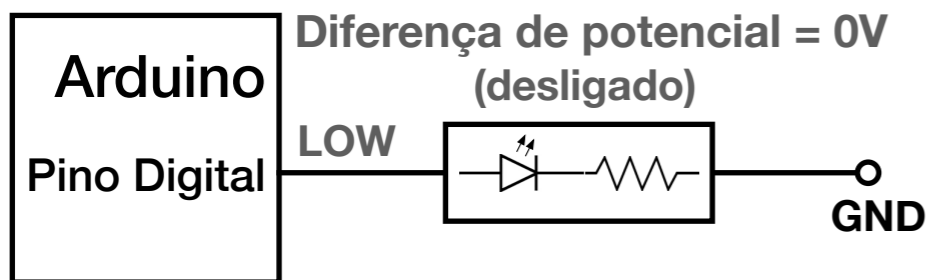
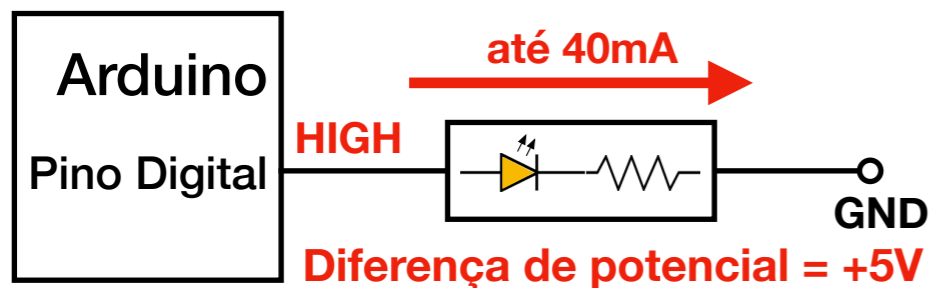
**LOW**

**0V** em qualquer Arduino

Requer configuração de pinMode: **OUTPUT**

Direção da corrente (quando há) depende da polaridade da **diferença de potencial** presente na carga (Ex:  $VCC - LOW = 5V$ )

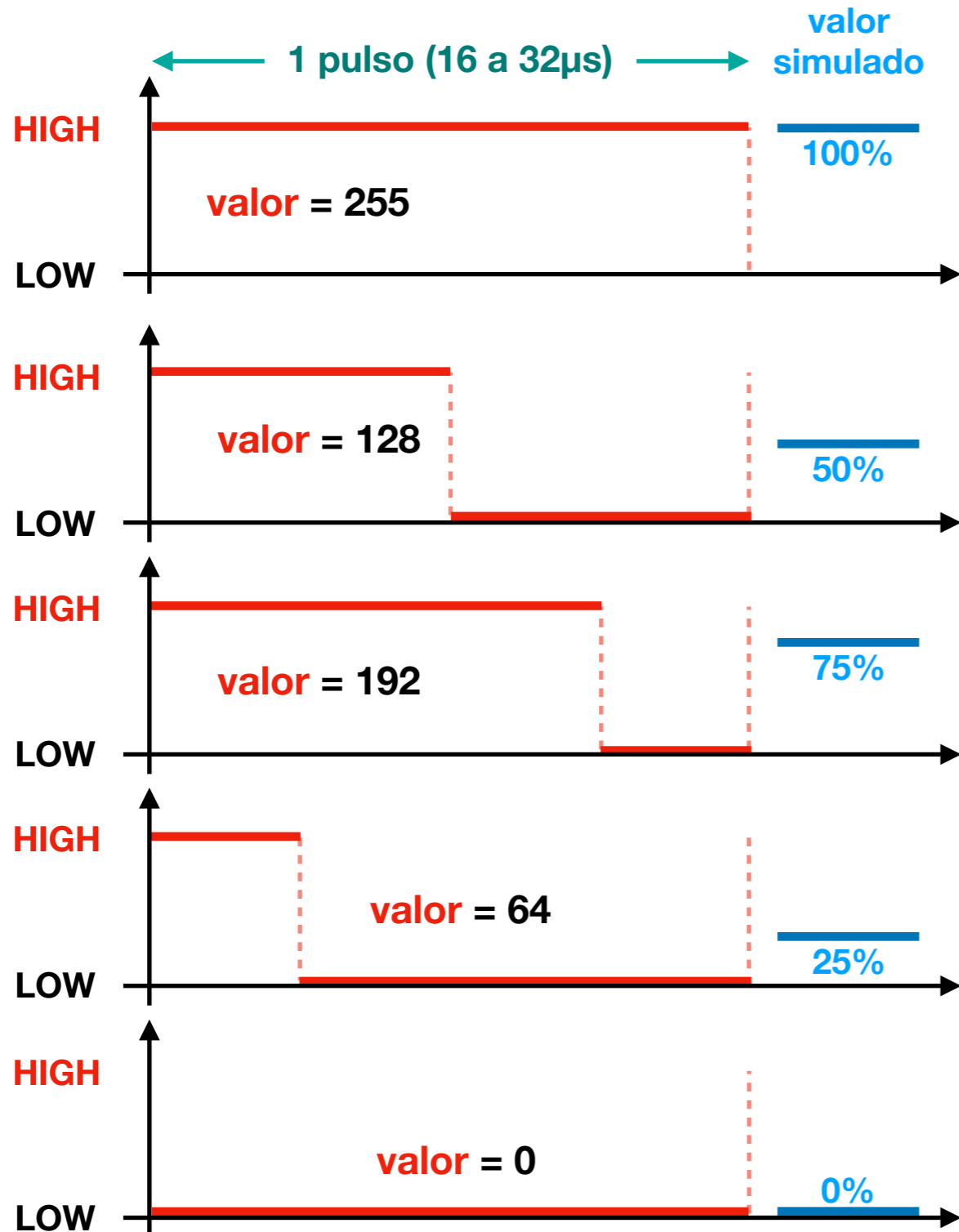
É necessário limitar a corrente (~40mA)



\* geração mais nova de LilyPad e ProMini



# analogWrite(pino, **valor**)



PWM (Pulse-Width Modulation) **simula** saída analógica (não é saída analógica verdadeira)

Acende (**HIGH**) e apaga (**LOW**) rapidamente; simula efeito analógico variando tempo em **HIGH** e **LOW**

Únicos valores produzidos são **VCC** (5V ou 3,3V) e **0V**

**Saída analógica verdadeira** requer circuito retificador (com capacitores)

Frequência do PWM em Arduino Uno/Nano

Pinos 3, 9, 10 e 11: **31250Hz**

Pinos 5 e 6: **62500Hz**

# int **valor** = analogRead(pino)

Valor lido é inteiro com resolução de 1/1024 da tensão de referência AREF

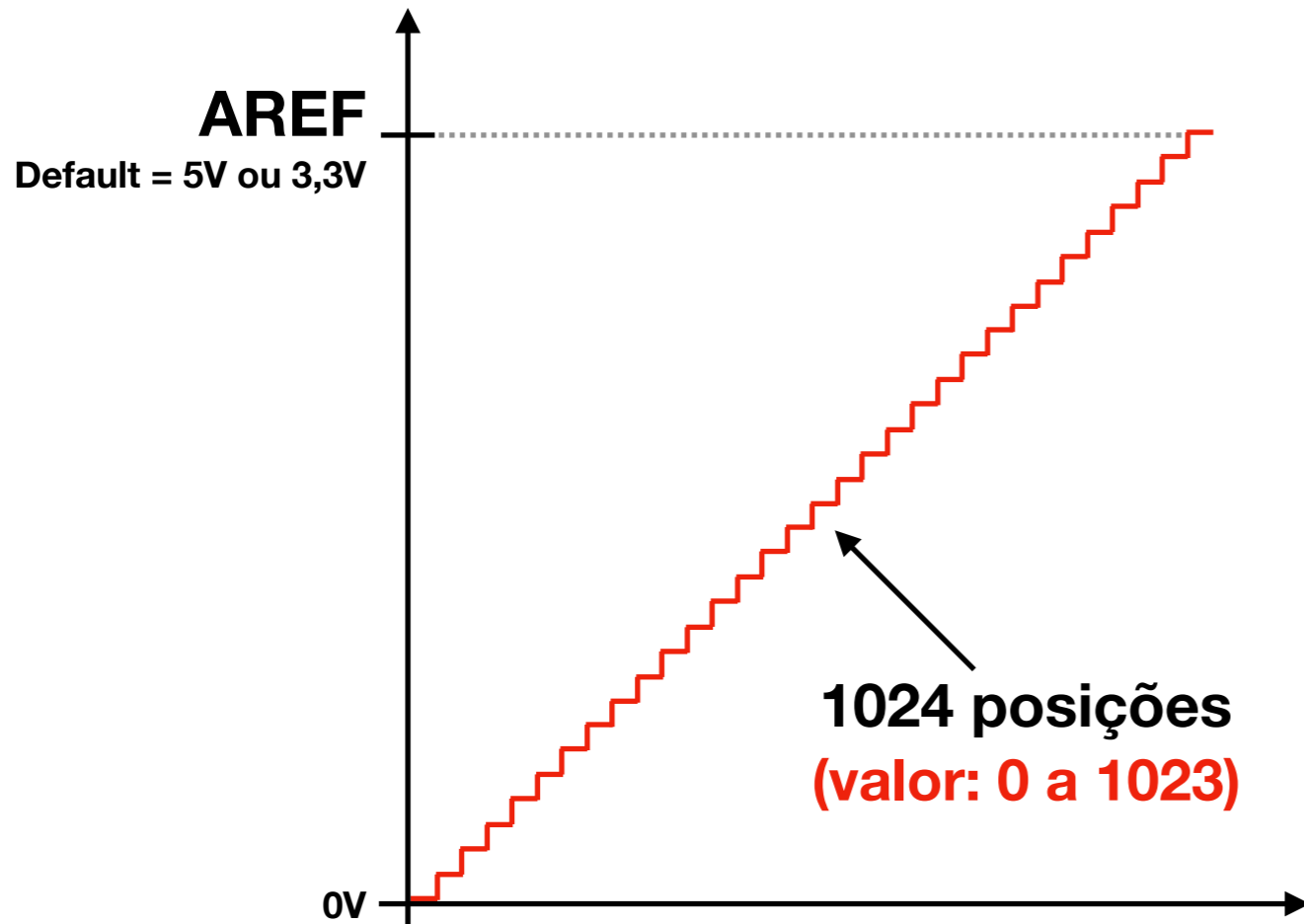
AREF é a **referência** de tensão máxima para leitura analógica.

Normalmente é usada a tensão VCC do Arduino (5V ou 3,3V)

Mas AREF pode ser alterada:

Via software, usando instrução **analogReference()**: 1,1V

Via hardware, aplicando tensão no **pino AREF**: tensão < 5V



Se AREF for 5V (Arduino Uno, Nano)

$$V = (\text{valor} / 1024) * 5V$$

Se AREF for 3,3V (Due, LilyPad, ProMini)

$$V = (\text{valor} / 1024) * 3,3V$$

Se AREF for 1,1V (Internal)

$$V = (\text{valor} / 1024) * 1,1V$$

AREF=VREF (<5V no pino AREF)

$$V = (\text{valor} / 1024) * VREF$$