

# WTF is an Arduino?

George Brindeiro





**NÃO LEMBRO DE MAIS NADA DA SEMANA PASSADA...**

# Potenciômetro

- Resistor ajustável
- Kit:  $2 \times 100 \text{ k}\Omega$
- Símbolo em esquemas:

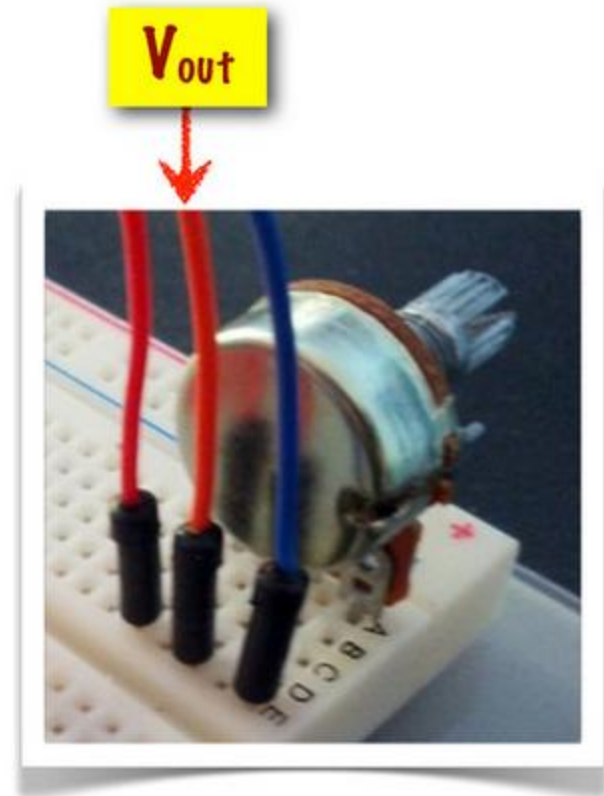


Fonte:

<https://speakerdeck.com/ramalho/arduino-101>

# Potenciômetro: como usar

- Ligar pinos laterais na alimentação
- Ligar pino central  $V_{out}$  em um pino de entrada analógico

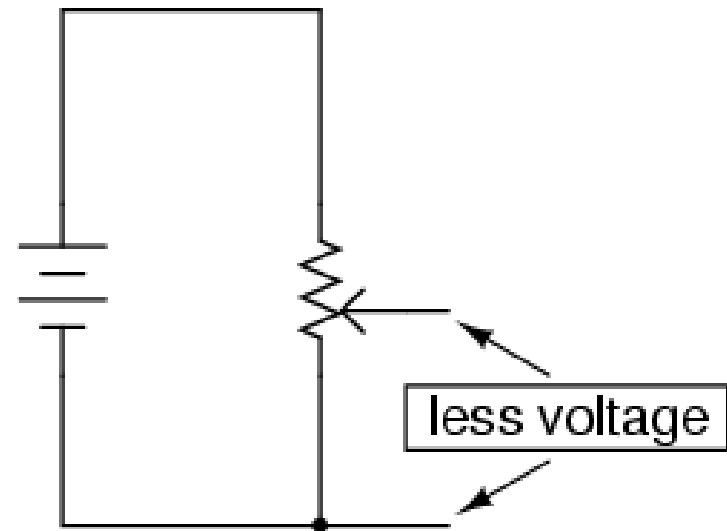
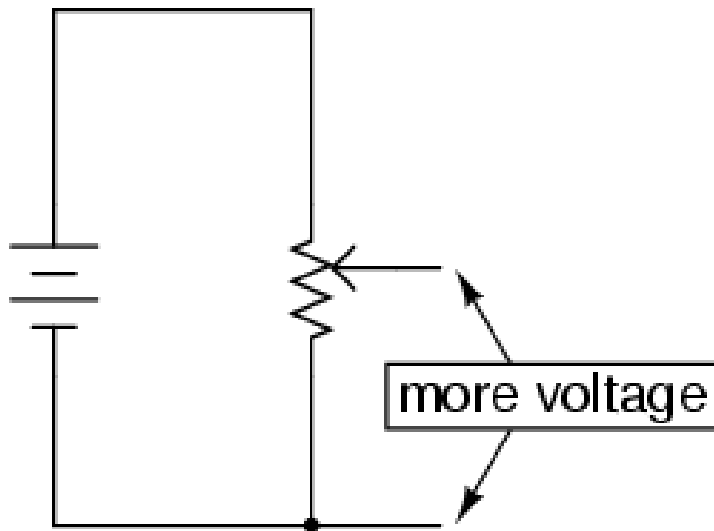


Fonte:

<https://speakerdeck.com/ramalho/arduino-101>

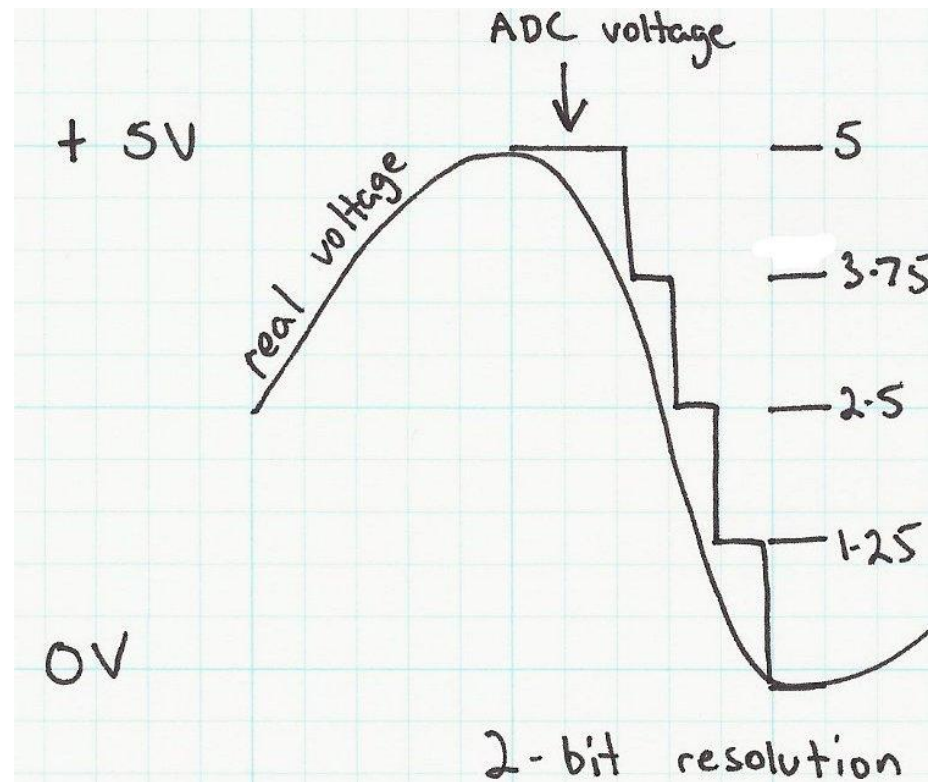
# Divisor de tensão: Potenciômetro

*Using a potentiometer as a variable voltage divider*



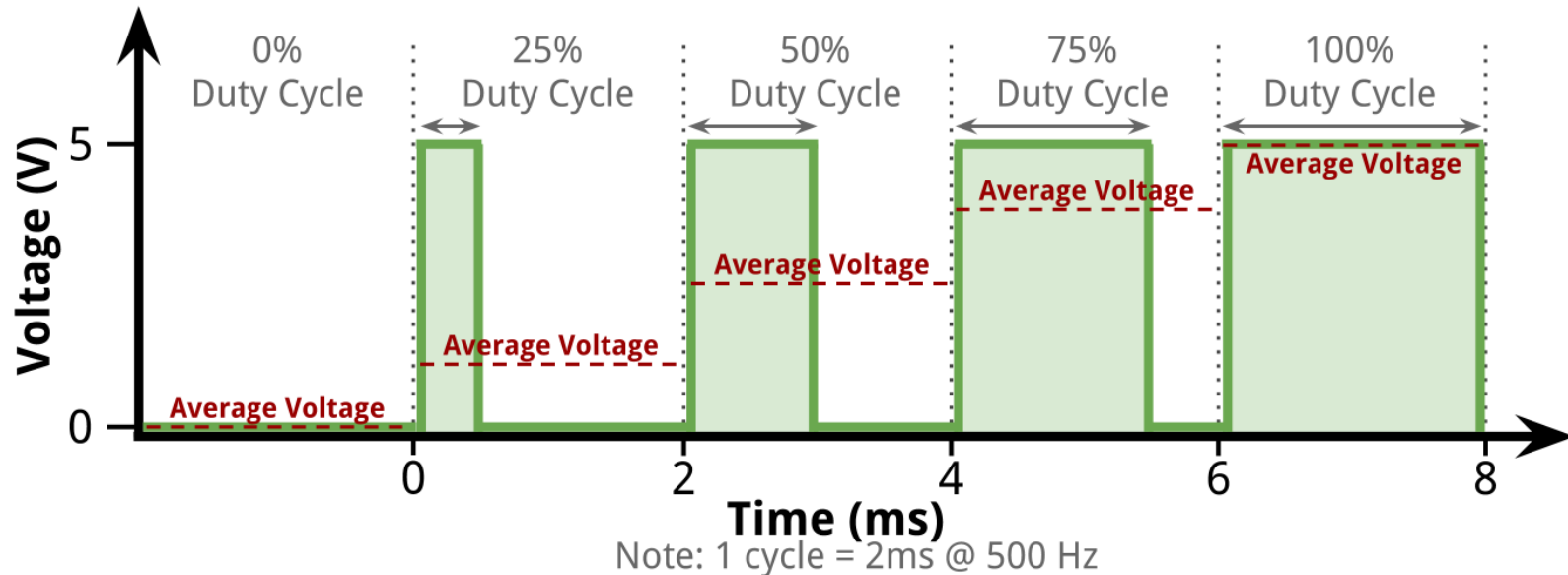
# ADC: Conversor Digital-Analógico

- VREF: máximo do ADC
- Resolução: tamanho da variável inteira (**digital**) usada pra representar sinal (**analógico**)
- # de Canais: quanto sinais dá pra ler com o mesmo circuito ADC



# PWM: Modulação por Largura de Pulso

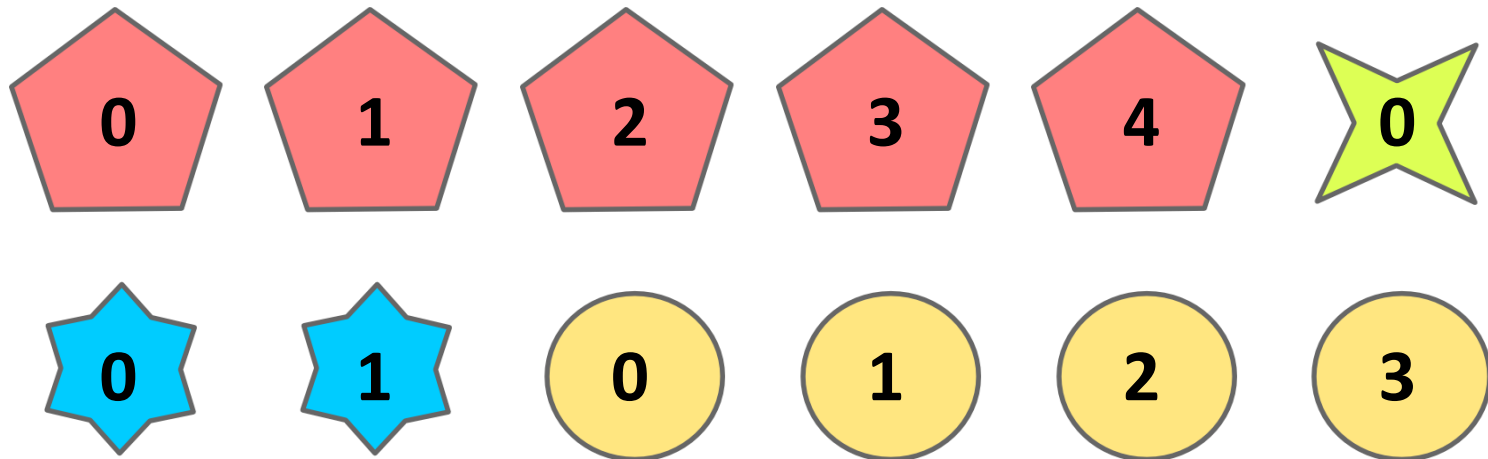
## Pulse Width Modulation Duty Cycles



- Contador de 8 bits: ciclos de 0 -> 255
- Circuitos “lentos”: só respondem ao valor médio!
- Outros casos pedem um capacitor... mas não vem ao caso

# Vetores

- Às vezes você quer guardar vários dados de forma fácil...
- Vetores permitem você fazer isso, acessando com índices
- Tamanho fixo, determinado quando você declara!





**WHILE**

**ANALOGREAD**

**MAP**

**DO/WHILE**



**CONTINUE**

**BREAK**



**FOR**

**ANALOGWRITE**

# ARDUINO REFERENCE

<http://arduino.cc/en/Reference/HomePage>

# Atualizando o sketchbook

- Arquivos do curso:  
<https://github.com/georgebrindeiro/wtf-is-an-arduino>
- Download ZIP e extrair em seguida
- File -> Preferences -> Sketchbook location
- Colocar a pasta extraída wtf-is-an-arduino/sketchbook

# Desafios: Conseguiram?

- Lesson2\_5\_UpDownBar + Lesson\_3\_2\_AnalogInOutSerial
  - Quero que o potenciômetro mude o brilho dos LEDs
  - E os botões digam quantos estão ligados/desligados!
- **Árvore de natal**
  - Quero três padrões diferentes de pisca
  - O potenciômetro deve mudar a velocidade de todos padrões
  - Os botões devem permitir escolher qual padrão é executado

## Entradas

Botões: pinos A1 e A2

Potenciômetro: pino A0

## Saídas

LEDs: pinos 11, 10, 9, 6, 5, 3

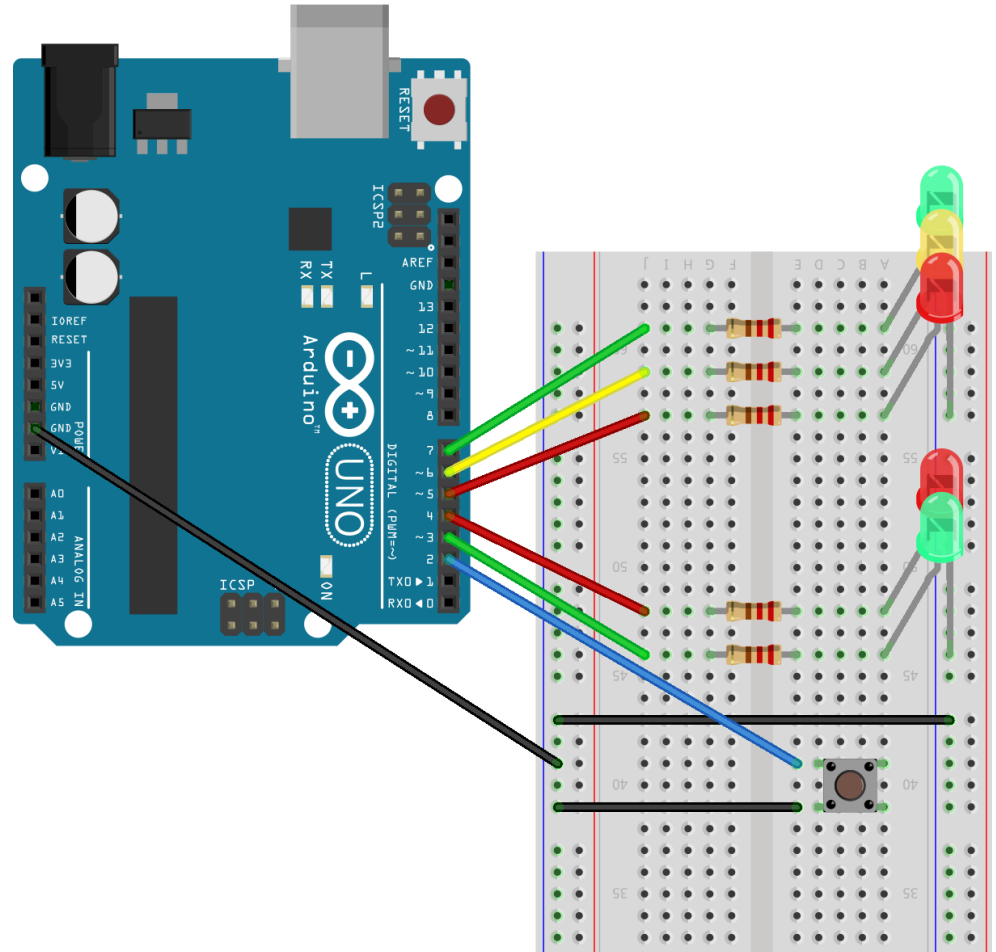
# 4

TEMPORIZAÇÃO E INTERRUPÇÕES:  
INDO ALÉM DO SETUP E LOOP



# Vamos montar!

- 6 resistores 330R
- 5 LEDs
- 1 botão
- Fios jumper
- Protoboard
- Arduino



# Lesson4\_1\_TestSemaphoreCircuit

- Esse código vai testar se está tudo ligado corretamente
- Vale a pena sempre pensar em códigos assim em projetos maiores. Isso impede que você fique na dúvida se o problema tá no circuito ou no código!
- Operador módulo (%) retorna o resto de uma divisão inteira
  - $7 / 4 = 1$  (divisão inteira) ou 3.75 (divisão ponto flutuante)
  - $7 \% 4 = 3$  (resto da divisão inteira)
- Isso permite criar ciclos de tamanho fixo de forma fácil!



# Como implementar a lógica?

- Quais situações existem?
- Quais eventos externos são relevantes para o sistema?
- Quais os critérios para a troca de comportamento?



WAIT



READY



GO!

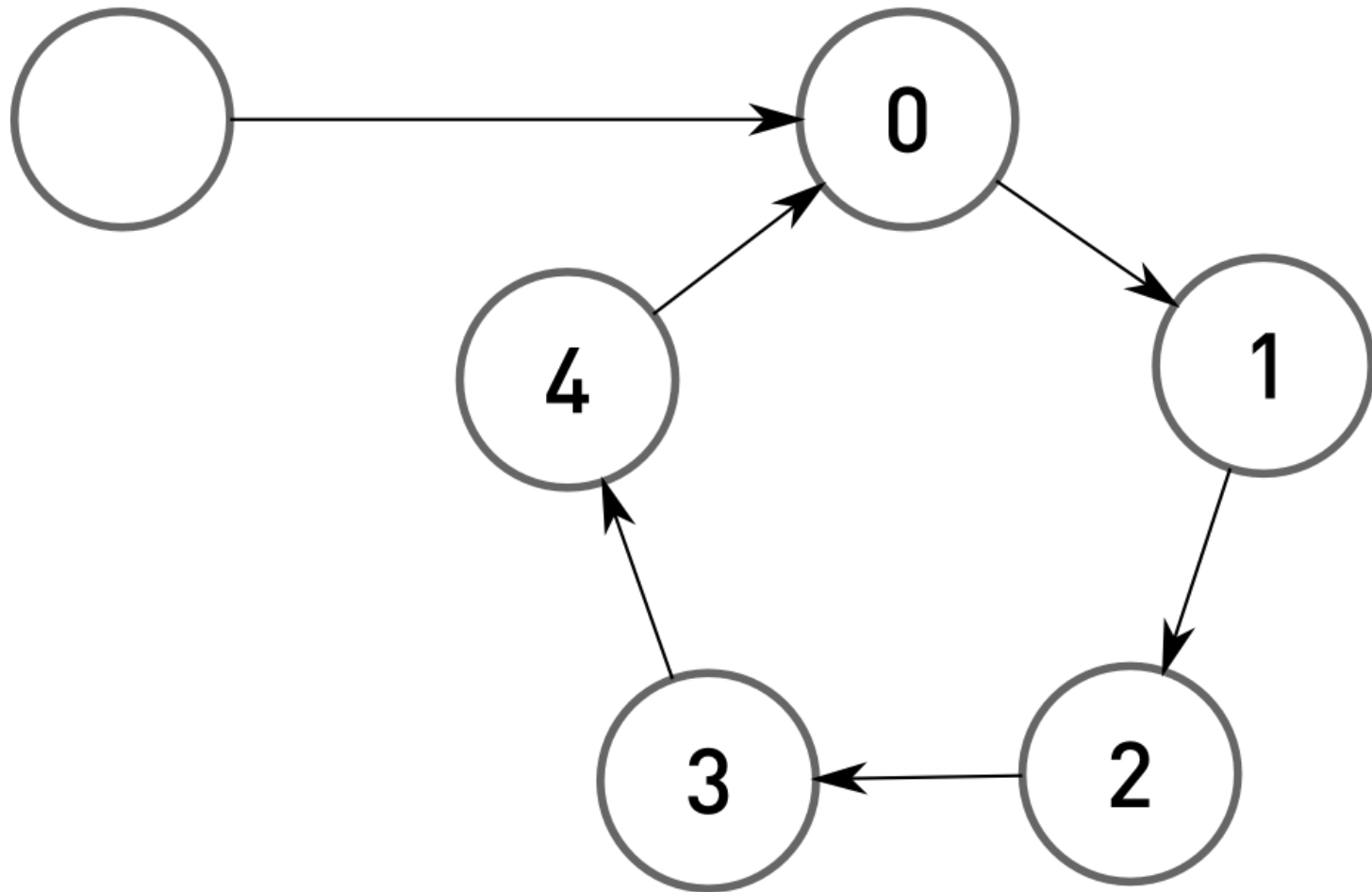


# Estados do sistema

Estado	Carros	Pedestres	Transição
0	VERDE	VERMELHO	pedWaiting && (t > t0)
1	AMARELO	VERMELHO	(t > t1)
2	VERMELHO	VERMELHO	(t > t2)
3	VERMELHO	VERDE	(t > t3)
4	VERMELHO	VERMELHO	(t > t4)

# Máquina de Estados

**setup**



# Lesson4\_2\_TestStateMachine

- Esse código vai mostrar como programar uma máquina de estados usando switch/case, para projetos com Arduino
- Para simplificar, vamos usar o botão para ativar a transição de um estado para outro. Nos próximos exemplos vamos incluir a temporização e a interação com o pedestre
- Usamos funções para deixar o código mais legível

# Funções

```
int fatorial(int n)
{
    int res = 1;
    for(int num = 1; num < n; num++)
        res = res*num;
    return res;
}
```

# Funções

```
int fatorial(int n)
{
    int res = 1;
    for(int num = 1; num < n; num++)
        res = res*num;
    return res;
}
```

## PARÂMETROS

Variáveis que são  
“enviadas” para uso  
dentro das funções

# Funções

## ESCOPO

Tempo/Espaço de vida das variáveis

```
int fatorial(int n)
{
    int res = 1;
    for(int num = 1; num < n; num++)
        res = res*num;
    return res;
}
```

# Funções

```
int fatorial(int n)
{
    int res = 1;
    for(int num = 1; num < n; num++)
        res = res*num;
    return res;
}
```

## RETORNO

Variáveis que são  
“retornadas” como  
resultado das funções



# Funções

```
int quemGanhou(int golsTime1, int golsTime2)
{
    if(golsTime1 > golsTime2)
        return 1;
    else if(golsTime1 < golsTime2)
        return 2;
    else
        return 0;
}
```

## RETORNO

Podemos passar quantas variáveis quisermos como parâmetro, mas só retornar uma como resultado

# Lesson4\_3\_TestOutputFunctions

- Esse código vai mostrar como usar funções para simplificar a leitura do que é feito em cada estado, além de evitar repetir trechos de código sem necessidade
- Poucas mudanças! Apenas adicionamos funções auxiliares para acender as luzes de acordo com a nossa lógica
- Quando começamos a fazer muita coisa em cada estado, demora pro Arduino verificar se o botão foi pressionado...
- Como lidar com essa situação?

An ornate, symmetrical decorative frame in a light color, featuring intricate scrollwork and floral motifs that enclose the central text.

Intermission

# Interrupção Externa

- Interatividade sem tomar todo o tempo
- O sinal em um pino desvia o fluxo do programa imediatamente!
- Não funciona em todos os pinos... (no UNO, pinos 2 e 3)

`attachInterrupt`(interrupt, ISR, mode)

- interrupt: 0 (pino 2) ou 1 (pino 3)
- ISR: nome da função que vai ser chamada
- mode: `LOW`, `CHANGE`, `RISING`, `FALLING`

# Lesson4\_4\_TestButtonInterrupt

- Esse código funciona exatamente da mesma forma que o primeiro, mas permite a resposta imediata ao botão
- Mesmo se adicionarmos um delay grande na função de cada estado, não temos problemas de interatividade
- Ainda existe o problema de “bounce”... como lidar?
- Exemplo de código comentado usando temporização



# Temporização

- Já ouviram falar em clock de um computador?
- Quantos GHz tem o seu computador?
- O que isso significa?



# Temporização

- O microcontrolador no “coração” do Arduino UNO, o ATmega328, usa um cristal de 16 MHz
- Dá pra gente usar isso pra contar o tempo, usando um contador interno de pulsos do clock!





# Que horas são?

- função `millis()`: quanto tempo desde o início do programa?
- unsigned long: 32 bits (faixa de 0 a 4 294 967 295)
- Contador volta para zero depois de aproximadamente 50 dias

`unsigned long millis()`

- retorna: quantos milissegundos desde o início do programa

**CUIDADO:** fazer contas de tempo com outros tipos inteiros pode te dar problemas. Evite dores de cabeça e use sempre unsigned long!

# Já chegamos? Já chegamos?

```
boolean isStateTimeOver(int state)
{
    unsigned long t = millis();

    if(t-t_init > t_state[state])
        return true;
    else
        return false;
}
```

# Lesson4\_5\_TestTiming

- Esse código vai fazer a transição de estados baseada **exclusivamente** em intervalos de tempo pré-estabelecidos
- O botão de pedestre está ativado, mas não influencia na lógica do programa! Vamos fazer isso no próximo passo 😊
- **MUITO IMPORTANTE:** guardar novo tempo de início  $t_{init}$  em cada transição! Sem isso, o sistema fica perdido!

# Lesson4\_6\_TestBlink

- Esse código vai adicionar apenas o LED piscando quando o tempo para pedestres está terminando
- Usando mais uma variável de estado (blink\_state) e um intervalo de tempo (t\_blink) podemos incorporar isso sem muito problema com um while
- **Notem:** agora o código entra uma só vez na função state\_4()

# Lesson4\_7\_Semaphore

- Agora é só integrar o botão de pedestre!
- Fazemos isso com uma variável booleana (true/false) que chamamos de `pedWaiting`, que muda para true quando o botão é pressionado, para sinalizar que há alguém esperando
- Essa variável é resetada para false quando o sinal fecha novamente, o que faz com que toques no meio do ciclo não levem o sinal a abrir novamente
- Usamos essa variável na condição de transição do estado 0!

# Próxima aula

- Conceber um projeto em grupo, com orientação
- Apresentação sobre sensores/atuadores
- Dicas de onde comprar e onde aprender mais
- Mais um projeto passo-a-passo, como o de hoje

MAIS ALGUMA IDÉIA?