WTF is an Arduino?

George Brindeiro

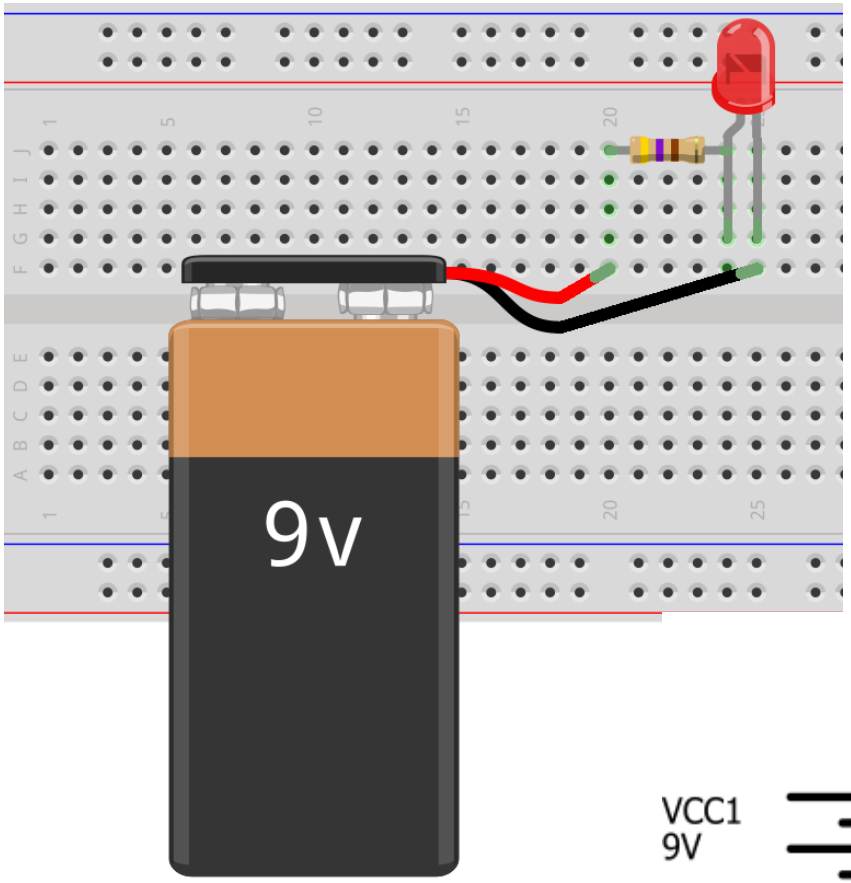
OVER DRIVE
ELETRÔNICA

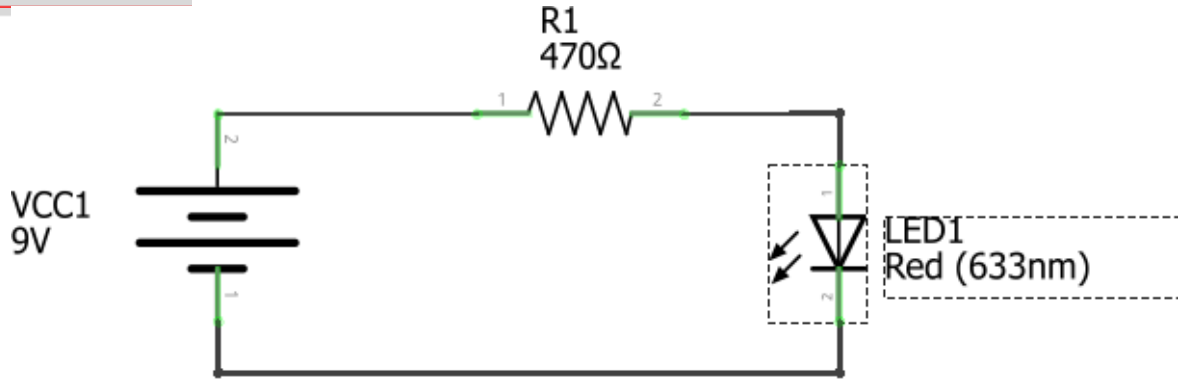
NÃO LEMBRO DE MAIS NADA DA SEMANA PASSADA...

CUIDADO

ENTRADA E SAÍDA DE VEÍCULOS

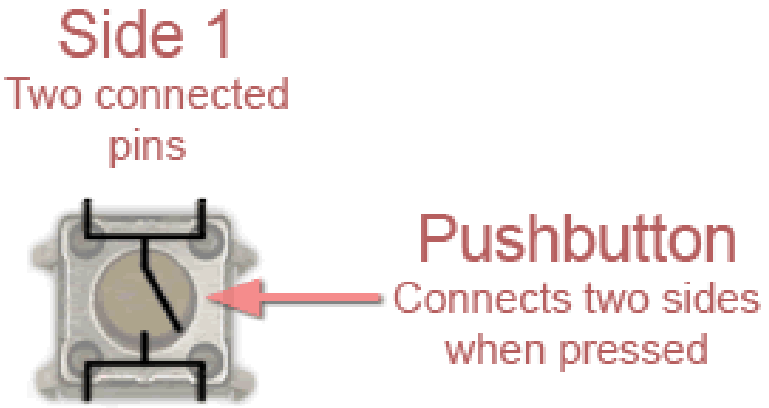
Circuitos básicos: LED e Resistor



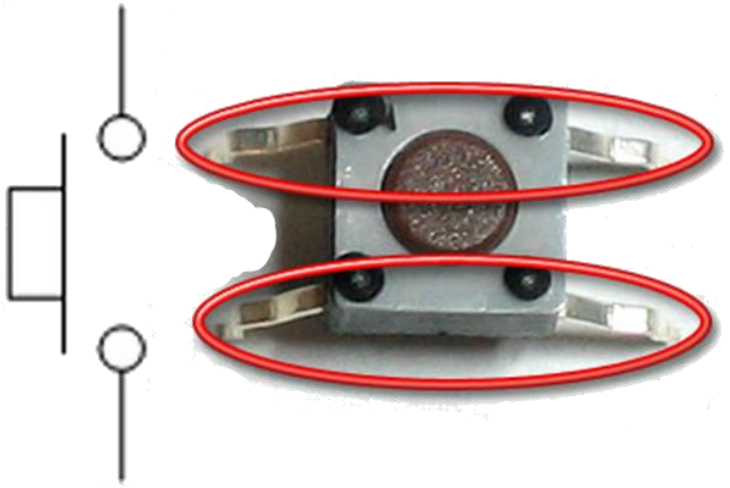


fritzing

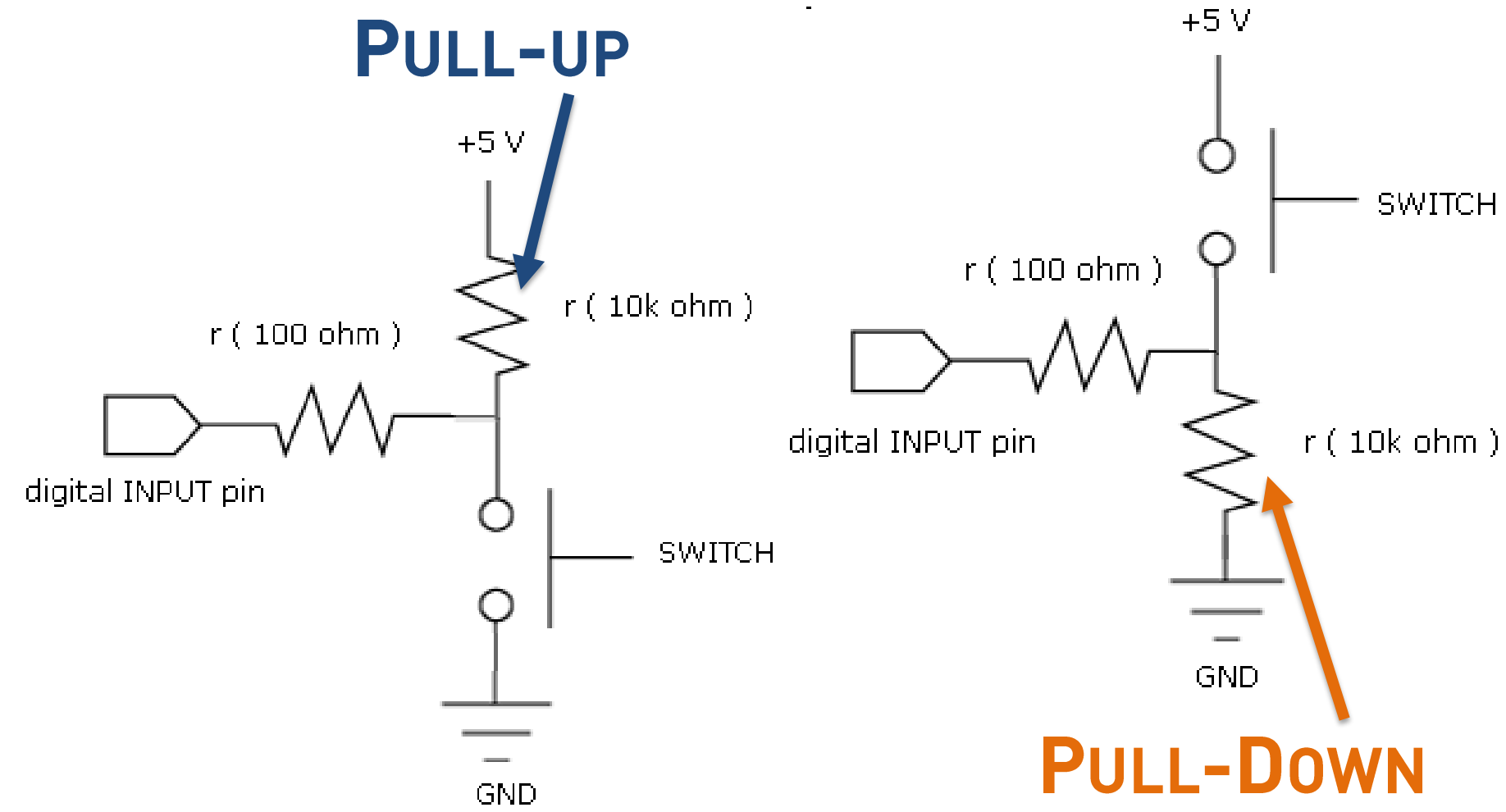
Botão “Pushbutton”



- Pares de pinos conectados (lados opostos)
- Normalmente aberto
- Apertar botão fecha circuito (conecta pinos eletricamente)



Truque: Pull-up ou Pull-down



Programando: Entrada/Saída Digital

Funções

pinMode(pin, mode)

digitalWrite(pin, level)

digitalRead(pin)

Argumentos

pin: 0-13 (analog: A0-A7)

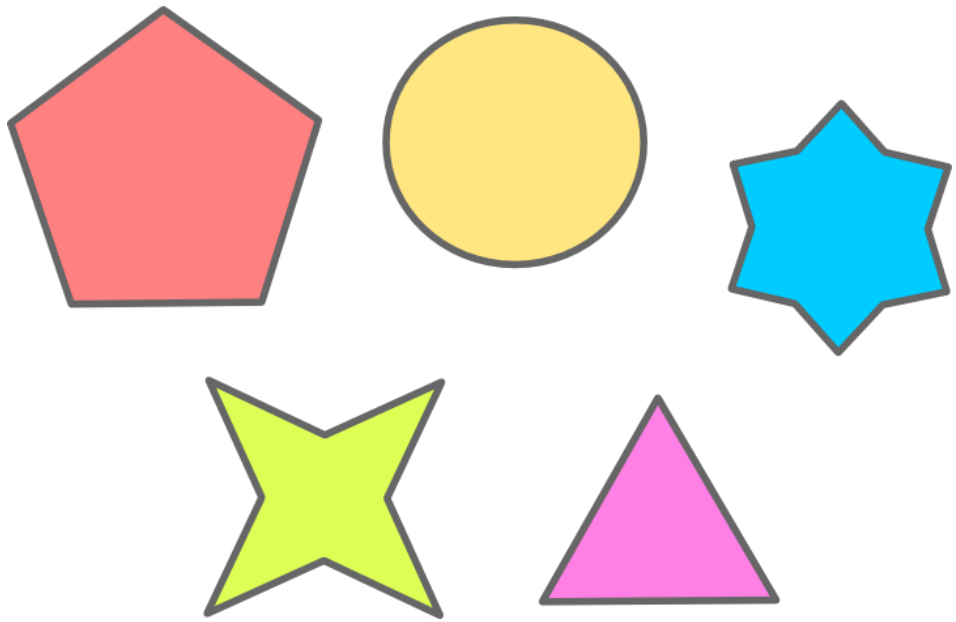
mode:

- INPUT para digitalRead
- INPUT_PULLUP para digitalRead com pull-up interno (> Arduino 1.0.5)
- OUTPUT para digitalWrite

level: HIGH (5V) ou LOW (GND)

Variáveis

- Às vezes você quer guardar um dado: número, texto, etc.
- Variáveis são "caixinhas" pra fazer isso



- char
- byte
- int
- unsigned int
- long
- unsigned long
- float
- double

http://arduino.cc/en/Reference/VariableDeclaration

IF

CONDIÇÃO

EXPRESSÕES

IF/ELSE

SWITCH/CASE

OPERADORES

Arduino Reference

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

Atualizando o sketchbook

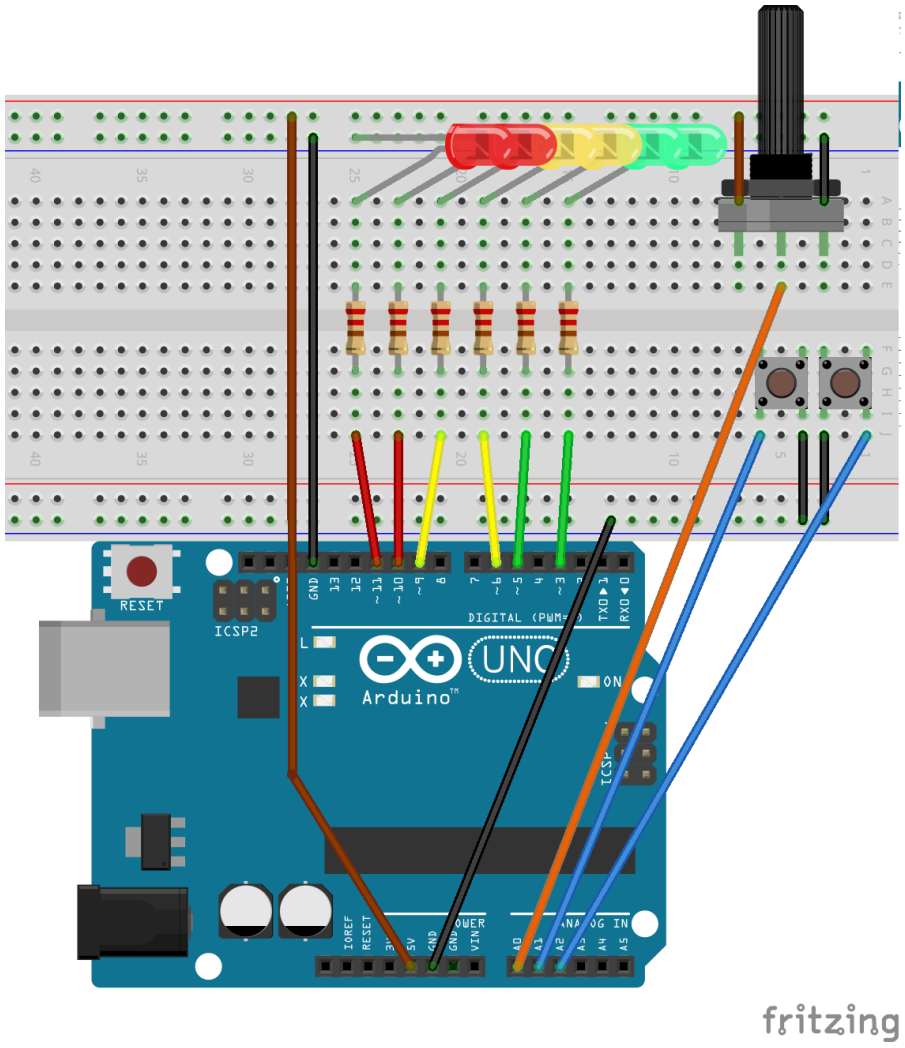
- Arquivos do curso: https://github.com/georgebrindeiro/wtf-is-an-arduino
- Download ZIP e extrair em seguida
- File -> Preferences -> Sketchbook location
- Colocar a pasta extraída wtf-is-an-arduino/sketchbook

3

Entrada e Saída Analógica: Superando a dicotomia HIGH/LOW

Vamos montar!

- 6 resistores 330R
- 6 LEDs
- 2 botões
- 1 potenciômetro
- Fios jumper
- Protoboard
- Arduino



Lesson3_1_AnalogInput

- Dêem upload no código e observem o comportamento
- Passo-a-passo do código: analogRead
- Modificação para olhar valores lidos na serial:
  - Adicionar no setup(): Serial.begin(9600);
  - Adicionar no loop(): Serial.println(sensorValue);
- Abram o Serial Monitor (lupa no canto superior direito)

Potenciômetro

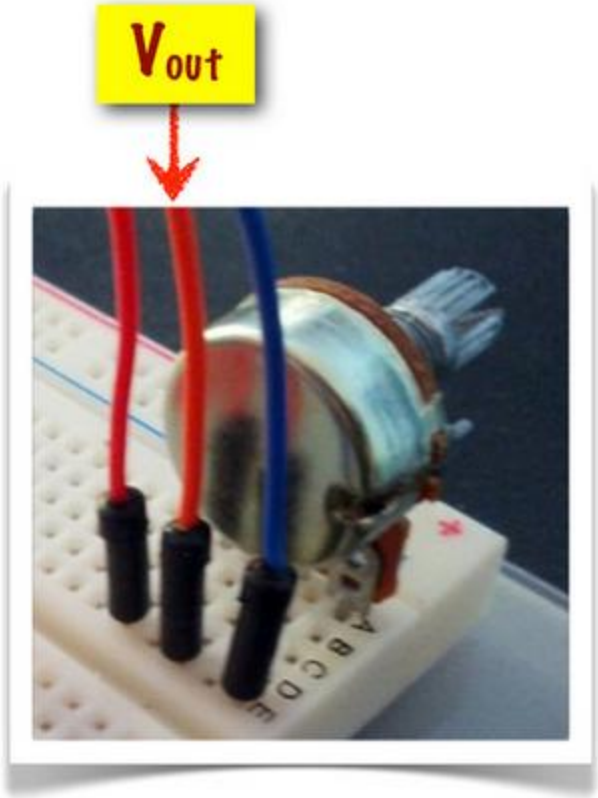
- Resistor ajustável
- Kit: 2 × 100 kΩ

- Símbolo em esquemas:

Fonte:
https://speakerdeck.com/ramalho/arduino-101

Potenciômetro: como usar

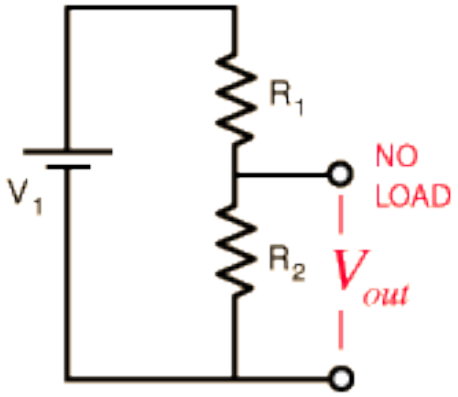
- Ligar pinos laterais na alimentação
- Ligar pino central V_{out} em um pino de entrada analógico



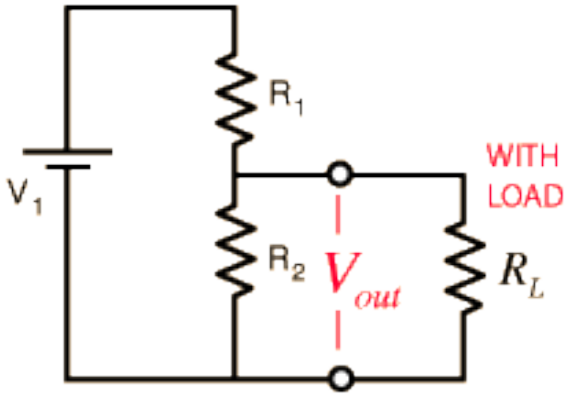
Fonte:
https://speakerdeck.com/ramalho/arduino-101

Divisor de tensão

OPEN CIRCUIT BEHAVIOR



BEHAVIOR UNDER LOAD



$$V_{out} = V_1 \frac{IR_2}{I(R_1 + R_2)} = \frac{V_1 R_2}{(R_1 + R_2)}$$

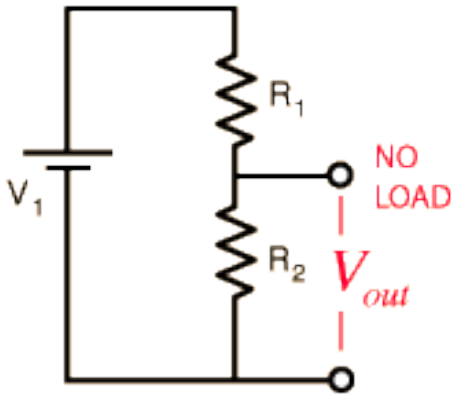
OUTPUT VOLTAGE UNDER "NO LOAD" CONDITION (open circuit)

OUTPUT VOLTAGE UNDER LOAD

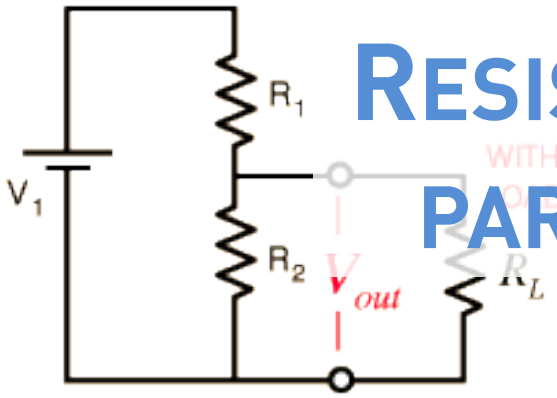
$$V_{out} = V_1 \frac{IR_2}{I(R_1 + R_2)} = \frac{V_1 (R_2 \parallel R_L)}{(R_1 + R_2 \parallel R_L)}$$

Divisor de tensão

OPEN CIRCUIT BEHAVIOR



BEHAVIOR UNDER LOAD



RESISTOR "EM PARALELO"

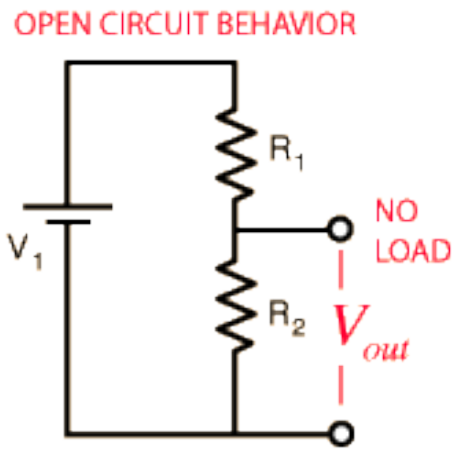
$$V_{out} = V_1 \frac{IR_2}{I(R_1 + R_2)} = \frac{V_1 R_2}{(R_1 + R_2)}$$

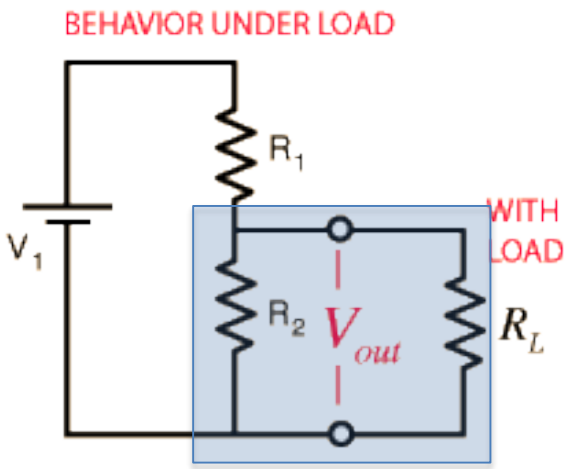
OUTPUT VOLTAGE UNDER "NO LOAD" CONDITION (open circuit)

OUTPUT VOLTAGE UNDER LOAD

$$V_{out} = V_1 \frac{IR_2}{I(R_1 + R_2)} = \frac{V_1 (R_2 \parallel R_L)}{(R_1 + R_2 \parallel R_L)}$$

Resistores em Paralelo



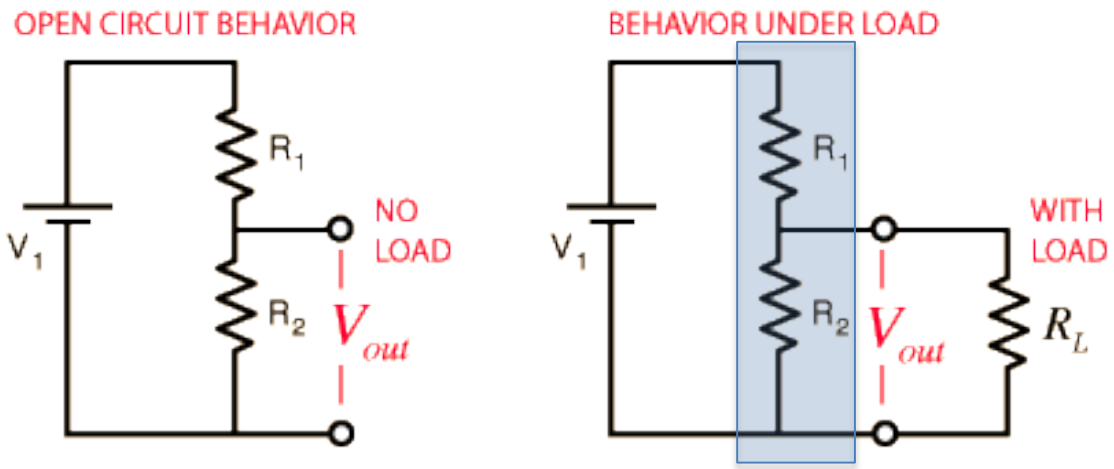


PARALELO

$$R_{eq} = \frac{R_a R_b}{R_a + R_b}$$

PARALELO = DIVIDE CORRENTE, MESMA TENSÃO

Resistores em Série



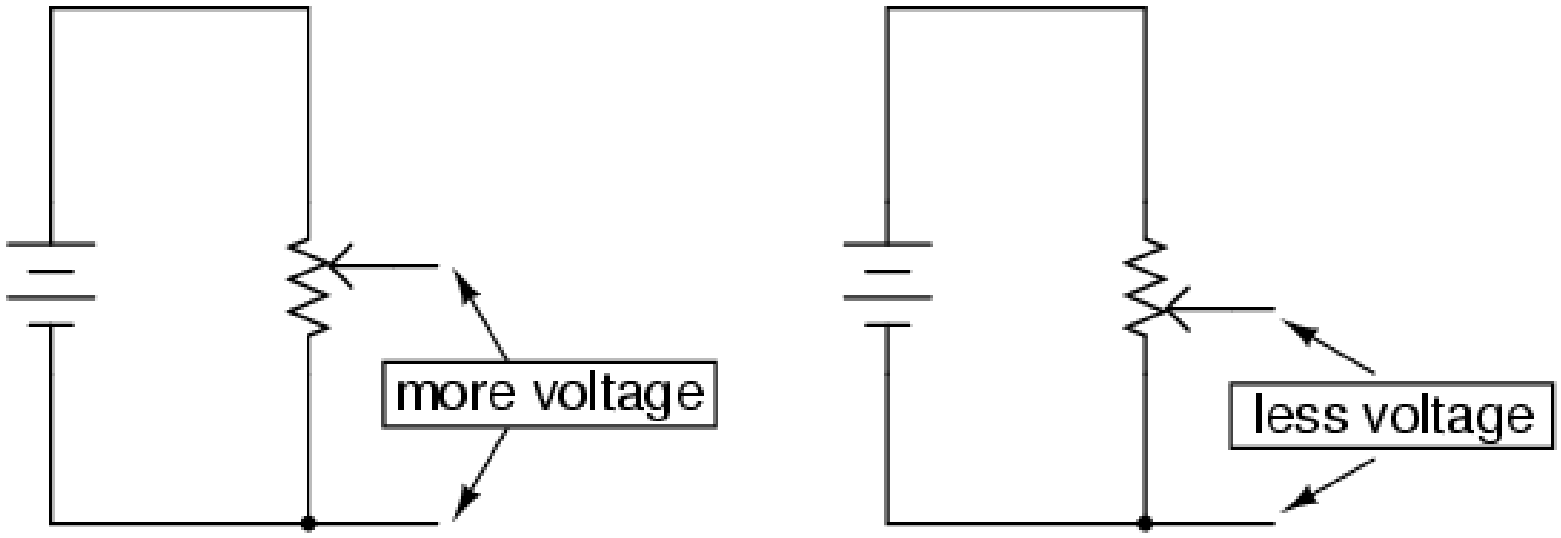
SÉRIE

$$R_{eq} = R_a + R_b$$

SÉRIE = DIVIDE TENSÃO, MESMA CORRENTE

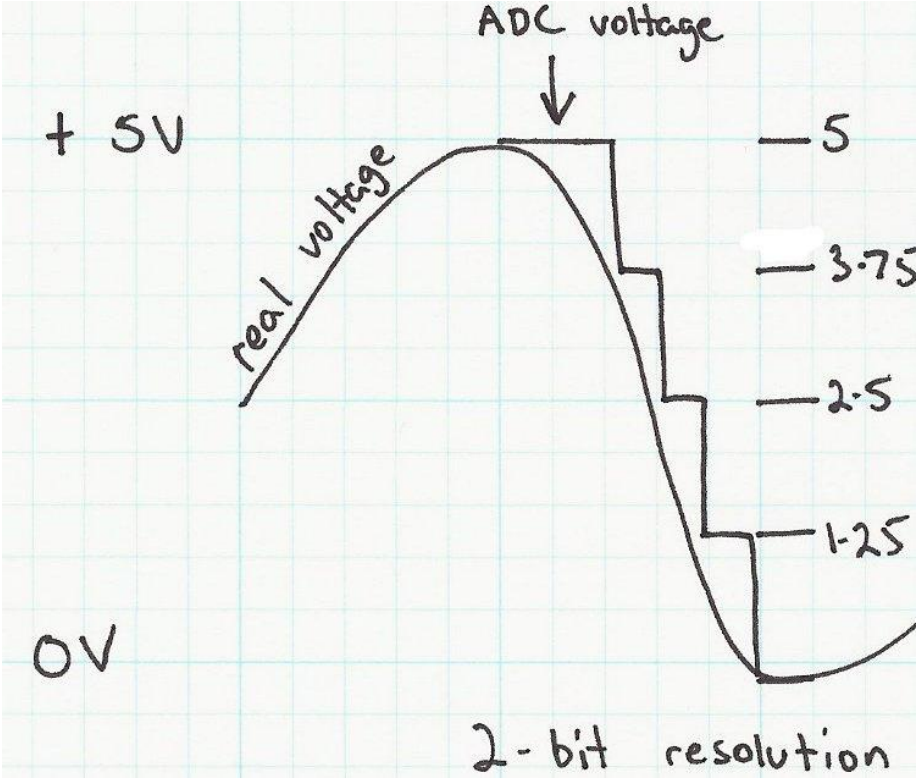
Divisor de tensão: Potenciômetro

Using a potentiometer as a variable voltage divider



ADC: Conversor Digital-Analógico

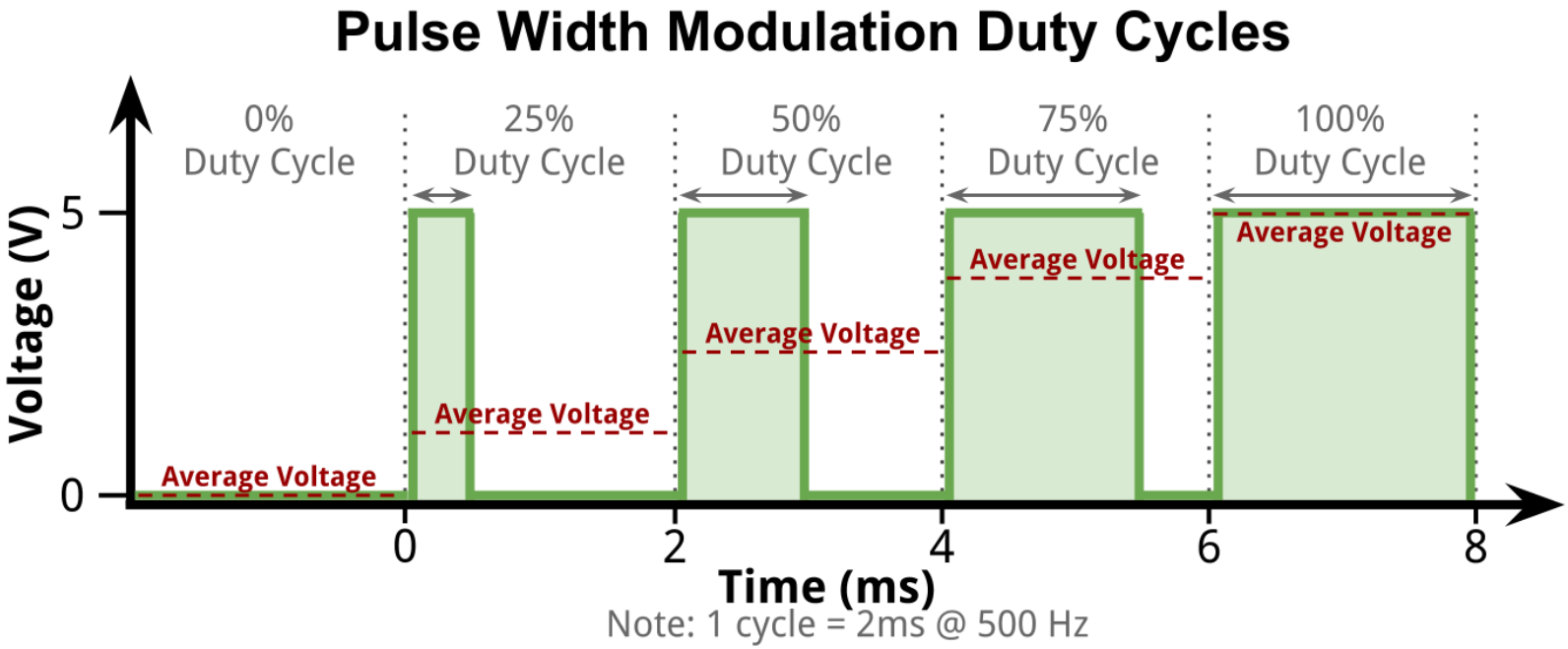
- VREF: máximo do ADC
- Resolução: tamanho da variável inteira (**digital**) usada pra representar sinal (**analógico**)
- # de Canais: quanto sinais dá pra ler com o mesmo circuito ADC



Lesson3_2_AnalogInOutSerial

- Dêem upload no código e observem o comportamento
- Passo-a-passo do código: analogWrite, map, const
- Abram o Serial Monitor e vejam o map funcionar!

PWM: Modulação por Largura de Pulso



- 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


Intermission

Lesson3_3_PotBar

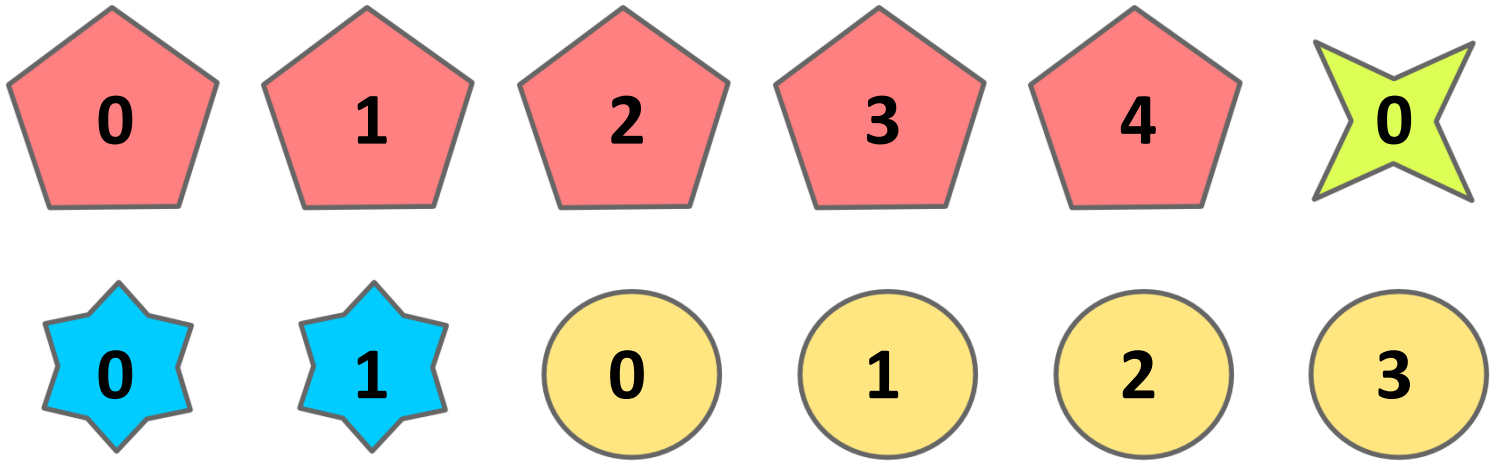
- Dêem upload no código e observem o comportamento
- Passo-a-passo do código: map em outro contexto, switch/case
- Funcionou como esperado? Se não, o que faltou?
- Mais um comando: break

Lesson3_4_PotLoopBar

- Dêem upload no código e observem o comportamento
- O que mudou pro exemplo anterior?
- Passo-a-passo do código: vetor, while, for

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!



http://arduino.cc/en/Reference/Array

while

- Enquanto condição for verdadeira, executa bloco de código

```
int sensorValue = 0;
while(sensorValue < 300)
{
    // código que vai ser executado se sensorValue < 300
    sensorValue = analogRead(A0);
}
```

do/while

- Igual ao while, só que testa condição no final. Ou seja: executa pelo menos uma vez, independente da condição.

```
int sensorValue;
do
{
    sensorValue = analogRead(A0);
} while(sensorValue > 300);
```

for

- Executa **initialização** e então repete bloco de código enquanto **condição** for verdadeira. Antes de testar a condição, executa a **expressão de fim de ciclo** (e.g. incremento/decremento)

```
for(int i = 0; i < 6; i++)
{
    // Imprime 0, 1, 2, 3, 4, 5 na porta serial
    Serial.println(i);
}
```

break

- Sai imediatamente do laço de repetição mais interno
- Sai imediatamente do switch/case

```
while(true)
{
    if(digitalRead(3))
        break;
}
```

continue

- Pula direto para a próxima execução do for, sem terminar o restante do bloco de código

```
for(int i = 0; i < 6; i++)
{
    Serial.println(i);
    if(i > 2) // Imprime 0, !, 1, !, 2, 3, 4, 5 na porta serial
        continue;
    Serial.println("!");
}
```

Lesson3_5_Fading

- Dêem upload no código e observem o comportamento
- Passo-a-passo do código
- Como fazer ficar mais rápido/devagar?

Desafios

- 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