FIAP GRADUAÇÃO

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Arquiteturas Disruptivas, IoT, Big Data e IA

PROF. ANTONIO SELVATICI

Versão 5– Fevereiro/2019

SHORT BIO

É engenheiro eletrônico formado pelo Instituto Tecnológico de Aeronáutica (ITA), com mestrado e doutorado pela Escola Politécnica (USP), e passagem pela Georgia Institute of Technology em Atlanta (EUA). Desde 2002, atua na indústria em projetos nas áreas de robótica, visão computacional e internet das coisas, aliando teoria e prática no desenvolvimento de soluções baseadas em Machine Learning, processamento paralelo e modelos probabilísticos. Desenvolveu projetos para Avibrás, IPT, CESP e Systax.

PROF. ANTONIO SELVATICI
profantonio.selvatici@fiap.com.br

INTERNET DAS COISAS

ARDUÍNO

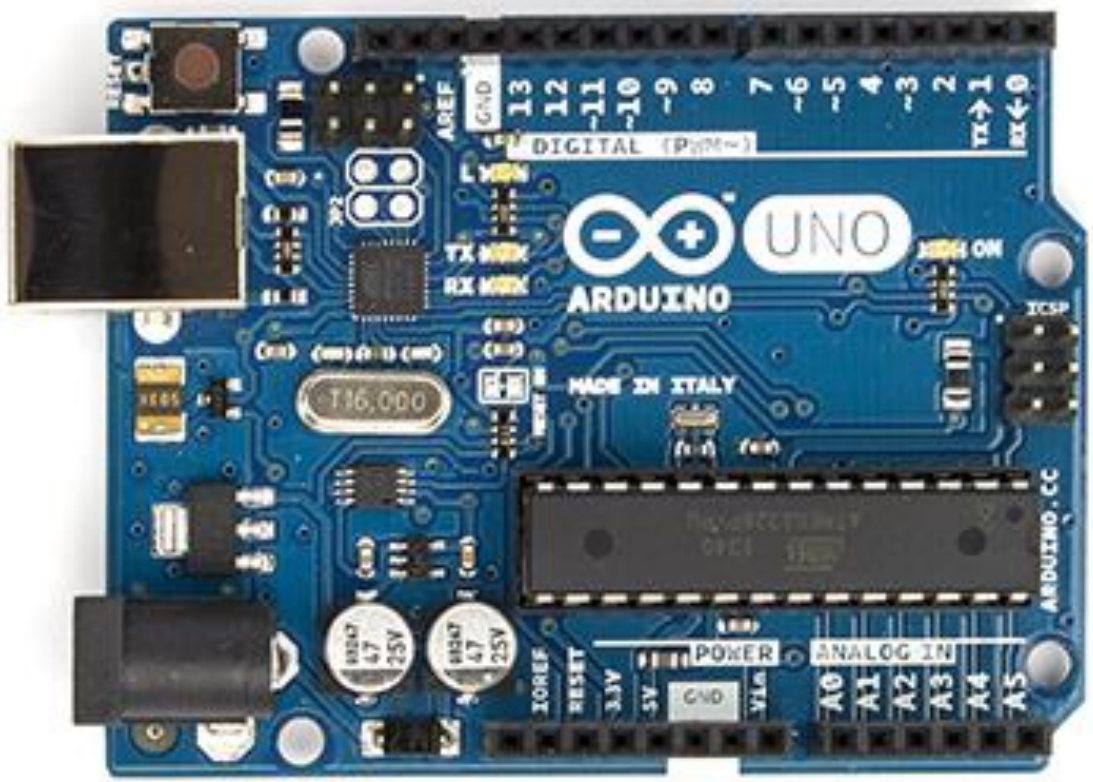
Introdução

- Arduino é uma plataforma de hardware para a rápida execução de projetos eletrônicos, possuindo um microcontrolador Atmel AVR com suporte de entrada/saída embutido
  - É um projeto *open source*, tanto no que tange ao hardware quanto ao software (ou seja, pode ser copiado)
  - Utiliza componentes de baixo custo
  - Emprega uma IDE de programação simplificada baseada em *Wiring*, que simplifica o processo de criação de projetos de C++
- Origem: criado por professores da Ivrea Interaction Design Institute para facilitar e baratear a criação de projetos pelos alunos
- Pode ser usado como um computador independente, ou estar conectado via USB no modo FTDI, sendo mapeado em uma porta serial.

ARDUINO UNO

TX- transmissao
RX- recepção

Arduino Uno



PRIMEIRO PROGRAMA

Hello World

- Formar grupos
- Conectar o Arduino ao computador pelo cabo USB
- Abrir o programa “Arduino IDE”
- Digitar o programa abaixo na janela que se abriu

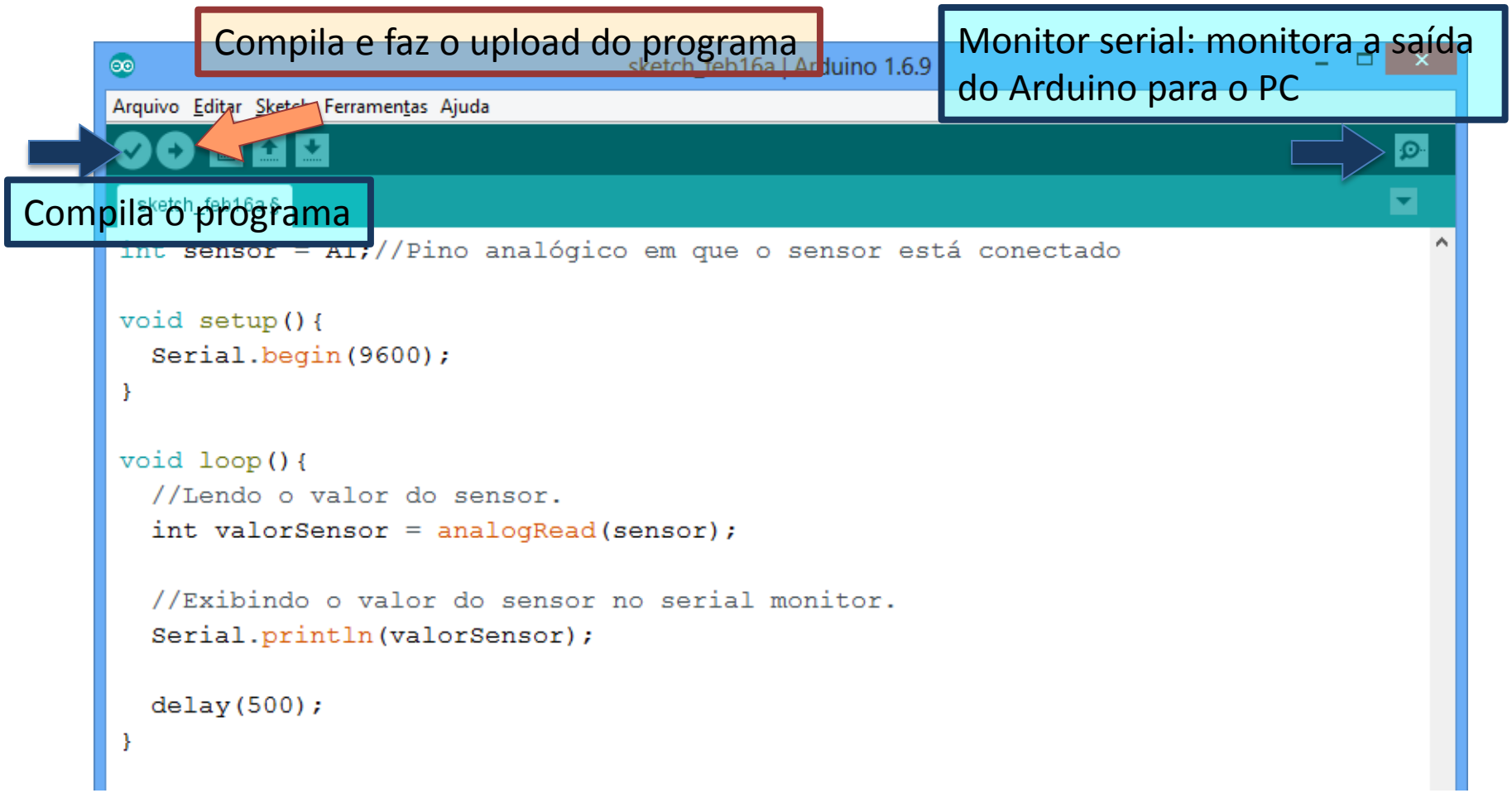
```
void setup() {
  Serial.begin(9600);
}
void loop() {
  Serial.println("Hello World!");
  delay(3000);
}
```

Taxa de comunicação

Serial- objeto

- Verificar e enviar o programa
- Abrir a janela Tools → Serial Monitor

Arduino IDE


Compila e faz o upload do programa
Monitor serial: monitora a saída do Arduino para o PC
Compila o programa
sketch_feb16a | Arduino 1.6.9
Arquivo Editar Sketch Ferramentas Ajuda
sketch_feb16a §
int sensor = A1;//Pino analógico em que o sensor está conectado
void setup(){
Serial.begin(9600);
}
void loop(){
//Lendo o valor do sensor.
int valorSensor = analogRead(sensor);
//Exibindo o valor do sensor no serial monitor.
Serial.println(valorSensor);
delay(500);
}

COMO FUNCIONA?

Hello World

- O Arduino possui uma tensão de operação de 5V, fornecida pela porta USB do computador
  - No caso de execução stand alone, deve ser fornecida alimentação de 7V a 12V na entrada de tensão DC (valores recomendados)
- Função `void setup();`
  - Chamada no início da execução do programa
  - Deve configurar os dispositivos a serem usados
- Função `void loop();`
  - Executada dentro do laço principal de execução
  - Executa enquanto a placa estiver ligada
- Classe Serial: representa a classe que se comunica com o computador hospedeiro através da porta USB/serial
- Função `delay(milissegundos)`: pausa a execução.

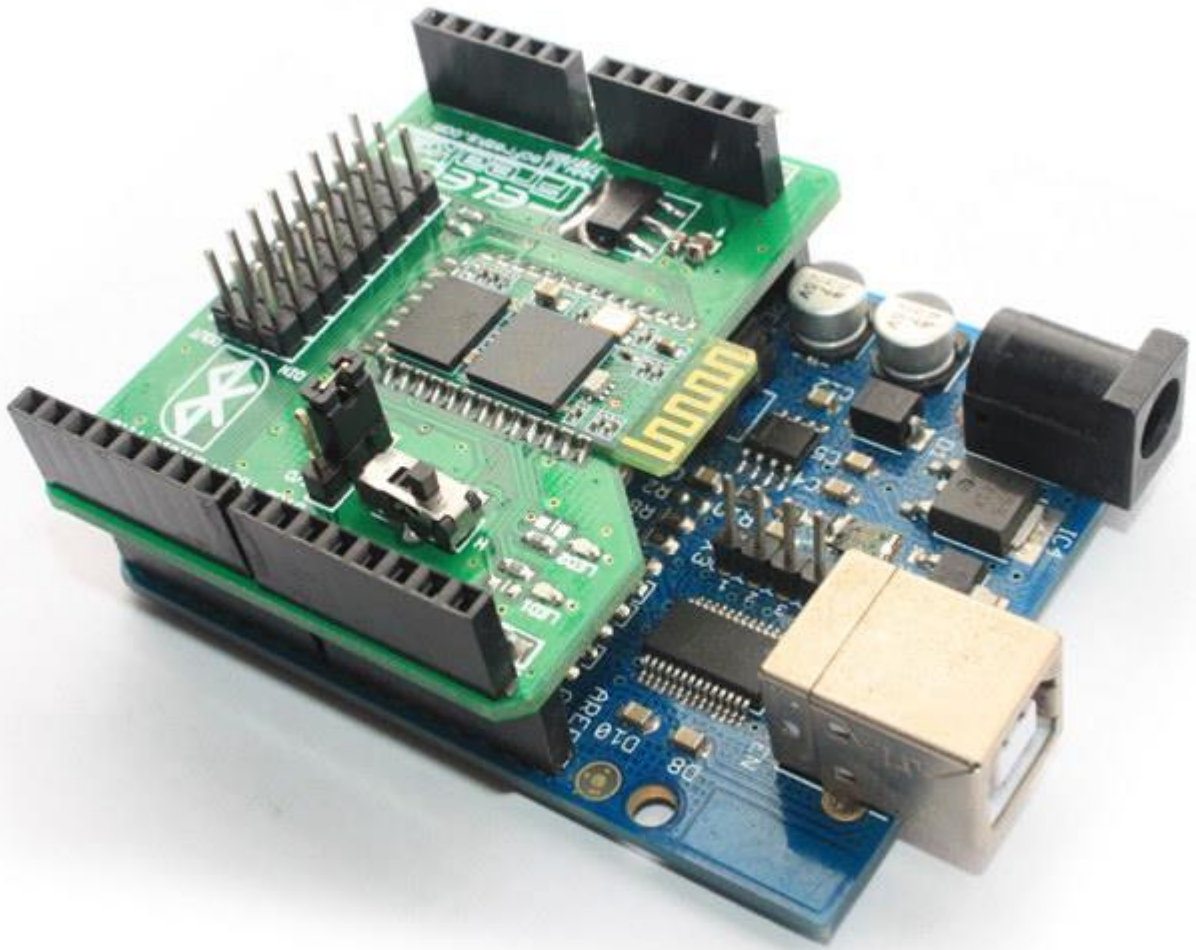
INTERAÇÃO COM DISPOSITIVOS

Sensores e atuadores

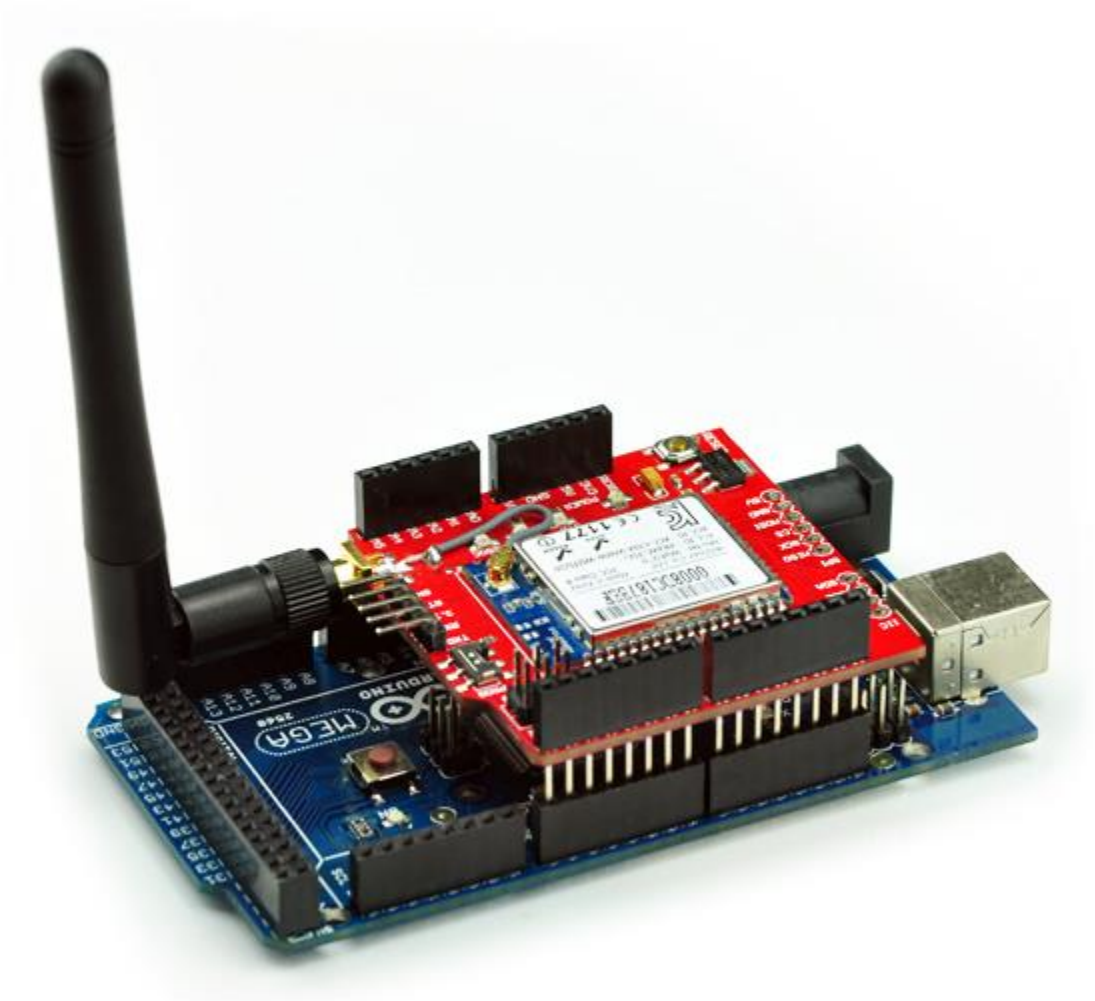
- A placa do Arduino tem como principal objetivo interagir com sensores e atuadores, servindo como controlador primário do sistema de automação, e, eventualmente, comunicando esses dados a um servidor.
  - Para que o Arduino leia dados de sensores, a placa dispõe de diversas *portas de entrada* de dados
  - Para que o Arduino envie comandos a atuadores, a placa dispõe de diversas *portas de saída* de dados
  - Para que o Arduino comunique-se numa rede de dispositivos de forma diferente do cabo USB, o projeto comporta a incorporação de *shields* à placa principal, estendendo suas funcionalidades.
  - Cada *shield* padrão tem uma biblioteca específica para sua utilização

ARDUINO COM SHIELD ETHERNET

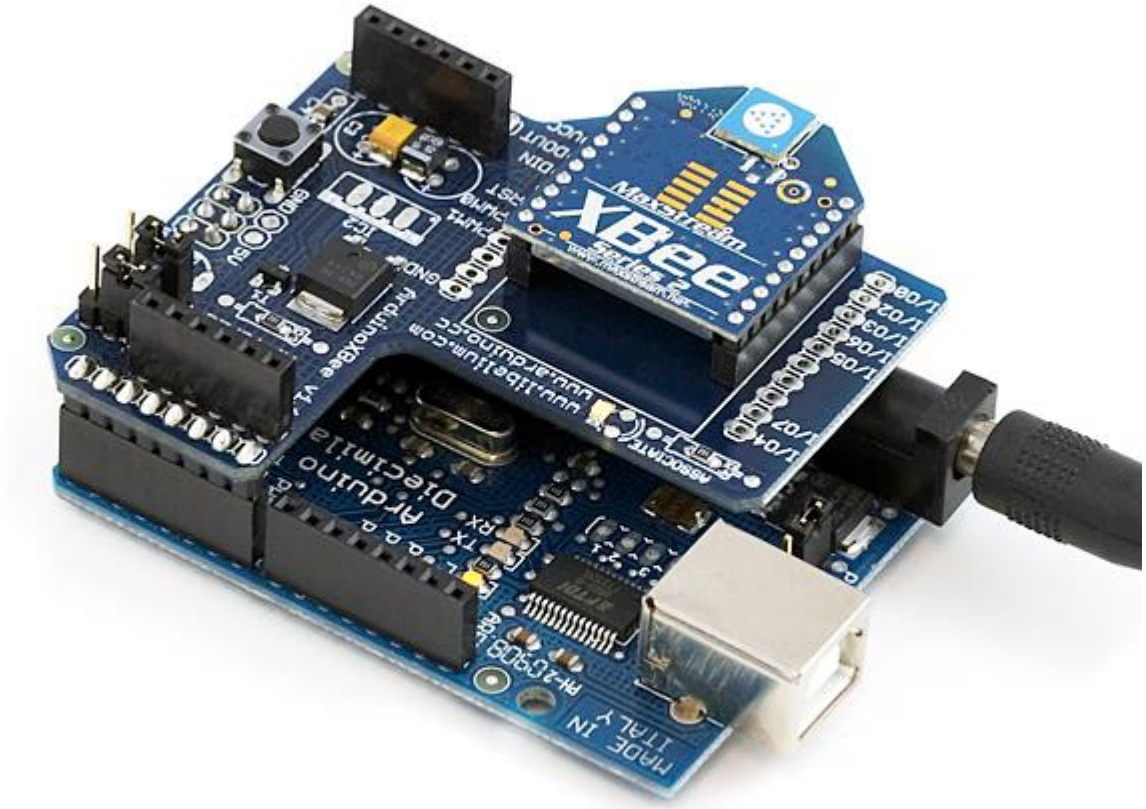
ARDUINO COM SHIELD BLUETOOTH



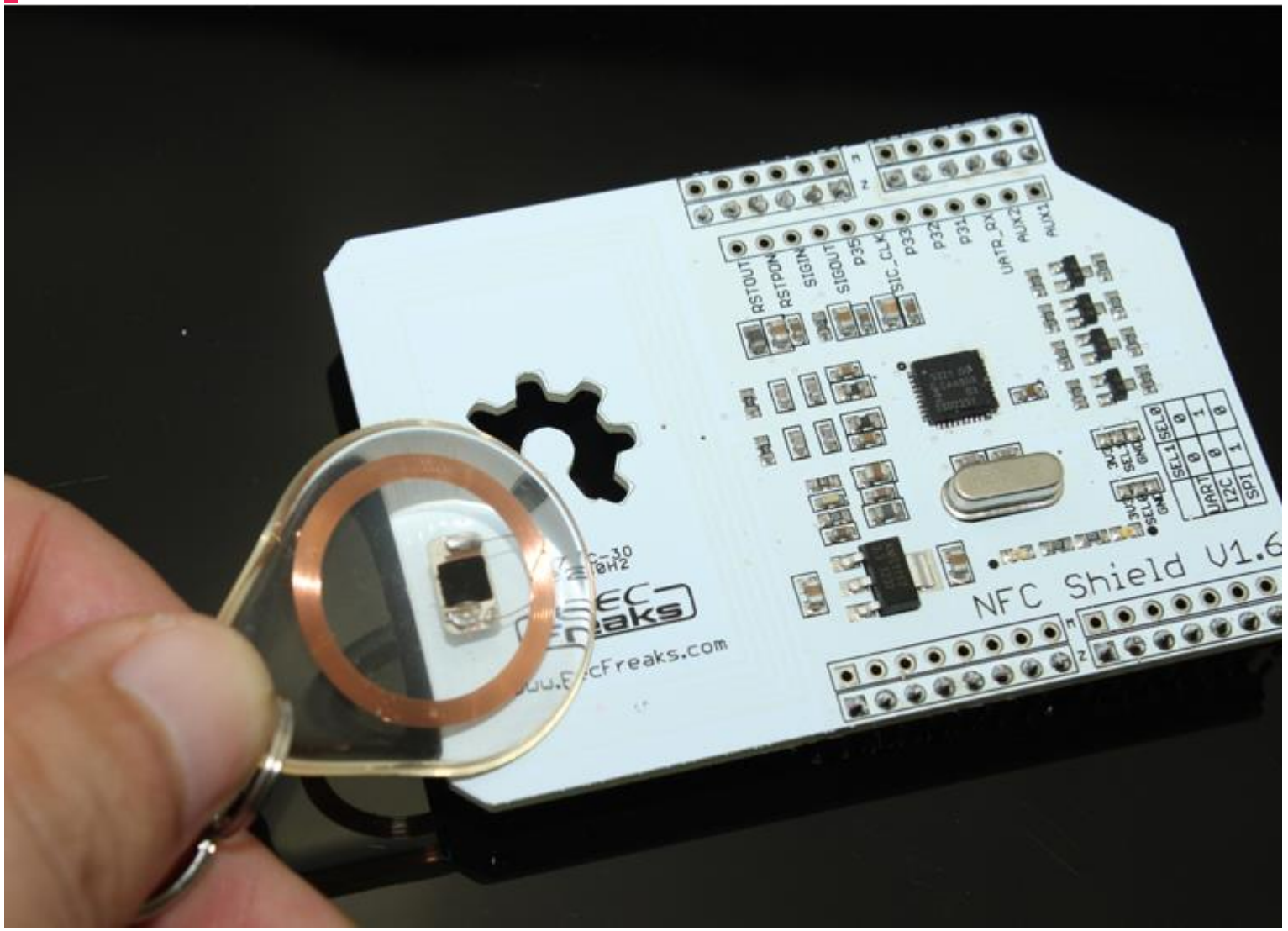
ARDUINO COM SHIELD WIFI



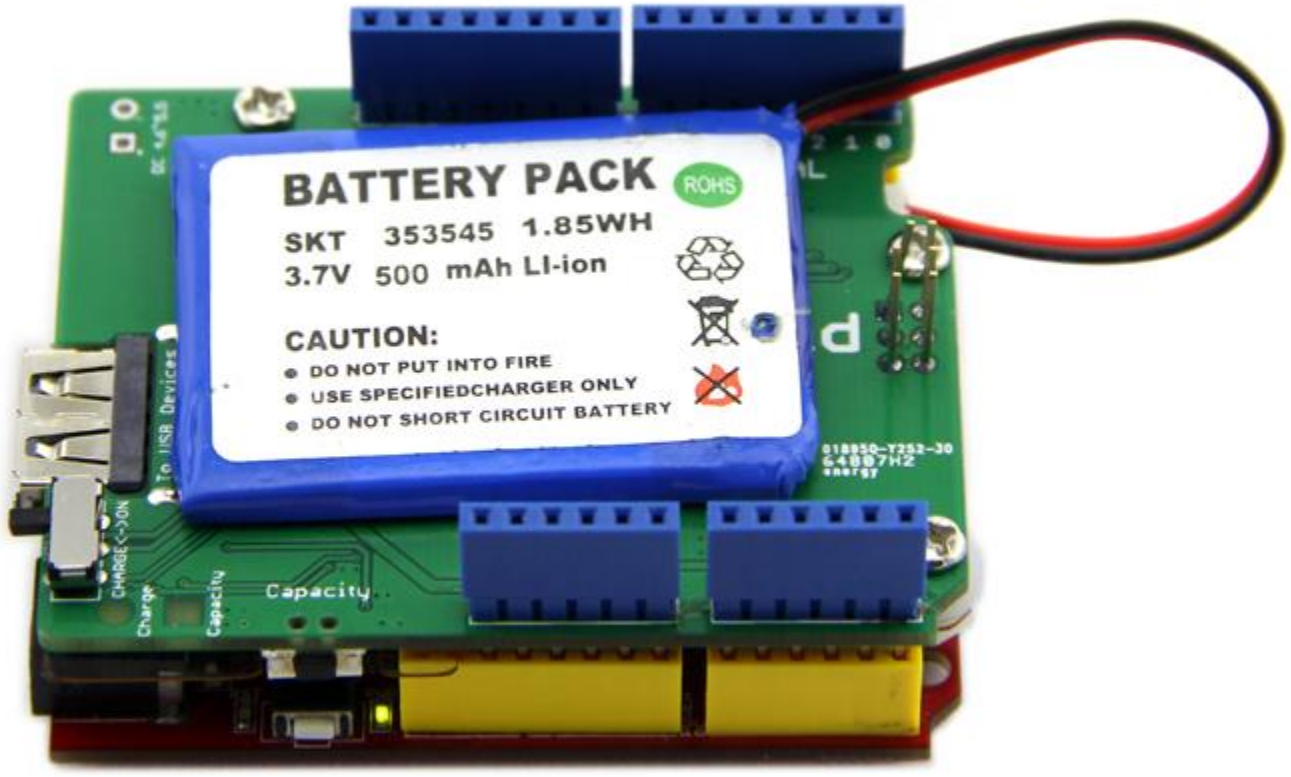
ARDUINO COM SHIELD ZIGBEE



SHIELD NFC


RSTOUT
RSTPDN
SIGIN
SIGOUT
P35
SIC_CLK
P33
P32
P31
UART_RX
AUX2
AUX1
SEL0
SEL1
UART
I2C
SPI
NFC Shield V1.6

ARDUINO COM SHIELD DE BATERIA



Sensores

Fazem a leitura de dados

- A leitura dos sensores é realizada pelas portas de entrada
  - São 6 portas de entrada analógicas...
  - Mais 14 portas de entrada ou saída digitais (GPIO)
- Sensor analógico: a saída do sensor é um nível de tensão, que deve ser capturado em uma das entradas analógicas e tem seu valor comparado com o valor de referência
  - No Arduino Uno, o padrão é 5V para essa referência
  - O valor retorna do varia de 0 (indicando 0V) a 1023 (indicando 5V)
  - Exemplos: sensor de temperatura, umidade, pressão do ar, luminosidade, etc.
- Sensor digital: a saída do sensor é um sinal ligado ou desligado, ou seja, um valor 0 ou 1 lógico
- A porta digital deve ser configurada para a leitura (modo input)
- Exemplo: apertar um botão

Tensão e corrente

- Trabalhar com o Arduíno requer o conhecimento de conceitos como tensão e corrente elétrica
  - Arduíno trabalha com lógica de 5V.
  - Cada porta de saída pode fornecer até 40 mA.
  - O que isso significa?
- A energia elétrica flui através do movimento das cargas elétricas livres, presentes nos materiais condutores elétricos, provocada por uma elevação do potencial elétrico em um ponto desse condutor.
  - Da mesma forma que uma pedra rola do alto da montanha para o vale , a carga elétrica livre tende a mover-se do ponto com maior potencial elétrico para o ponto de menor potencial
  - À diferença no potencial elétrico de um ponto a outro chamamos **diferença de potencial** ou **tensão elétrica**
  - À vazão de cargas elétricas que se movimentam dentro do condutor em um certo ponto chamamos **corrente elétrica**

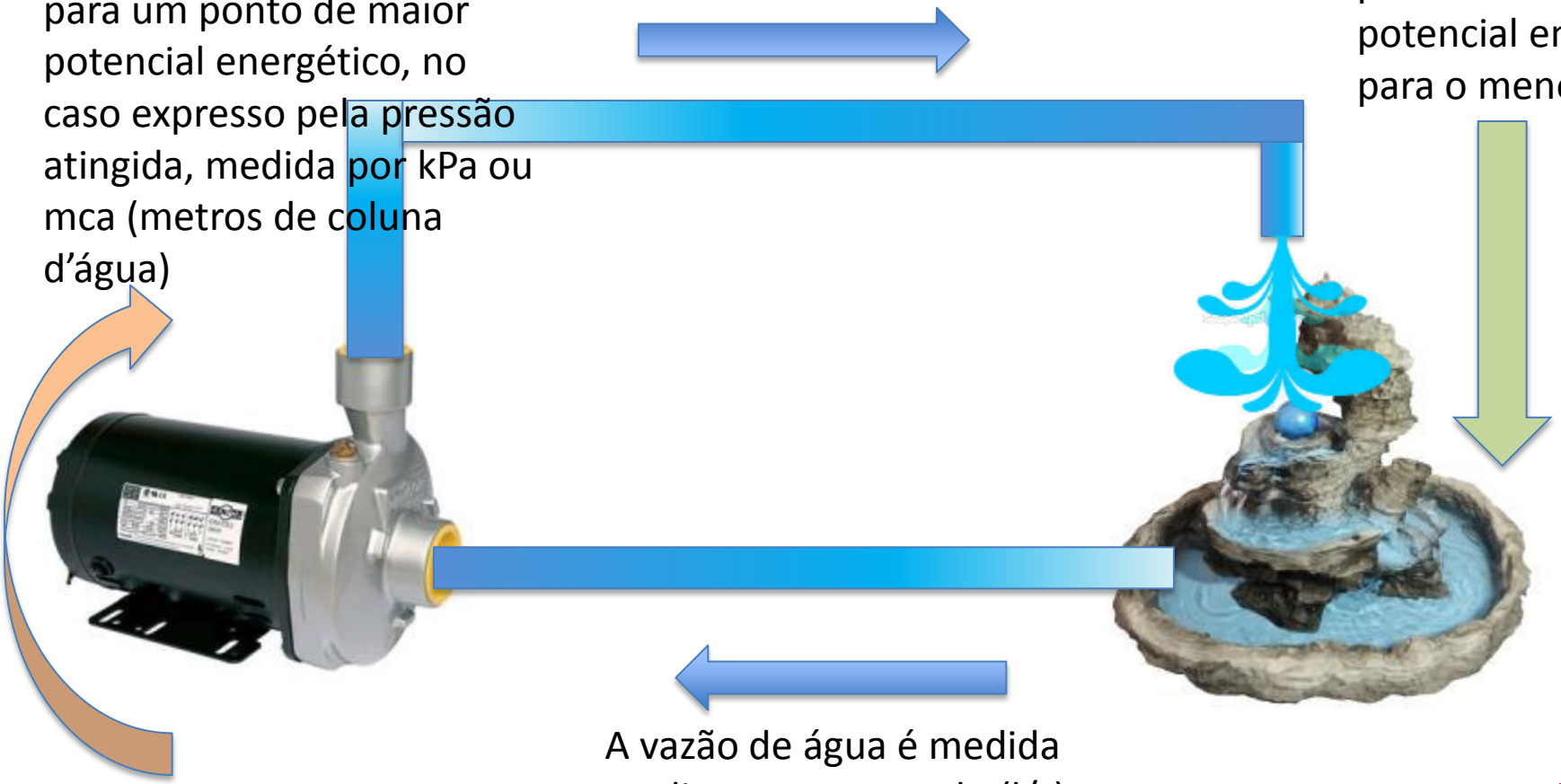
Circuitos elétricos

- Para usarmos a energia elétrica, precisamos criar uma diferença de potencial, que por sua vez irá impor uma corrente elétrica a um material condutor
- A maneira que usamos a energia elétrica é na forma de um circuito elétrico, onde a corrente elétrica está sempre circulando devido a um fornecimento ininterrupto de tensão elétrica.
- Vamos comparar o circuito elétrico a uma fonte que está sempre jorrando água

Circuito elétrico: analogia com uma fonte

A bomba d´água usa energia externa para levar a água para um ponto de maior potencial energético, no caso expresso pela pressão atingida, medida por kPa ou mca (metros de coluna d'água)

A água flui naturalmente do ponto de maior potencial enérgico para o menor

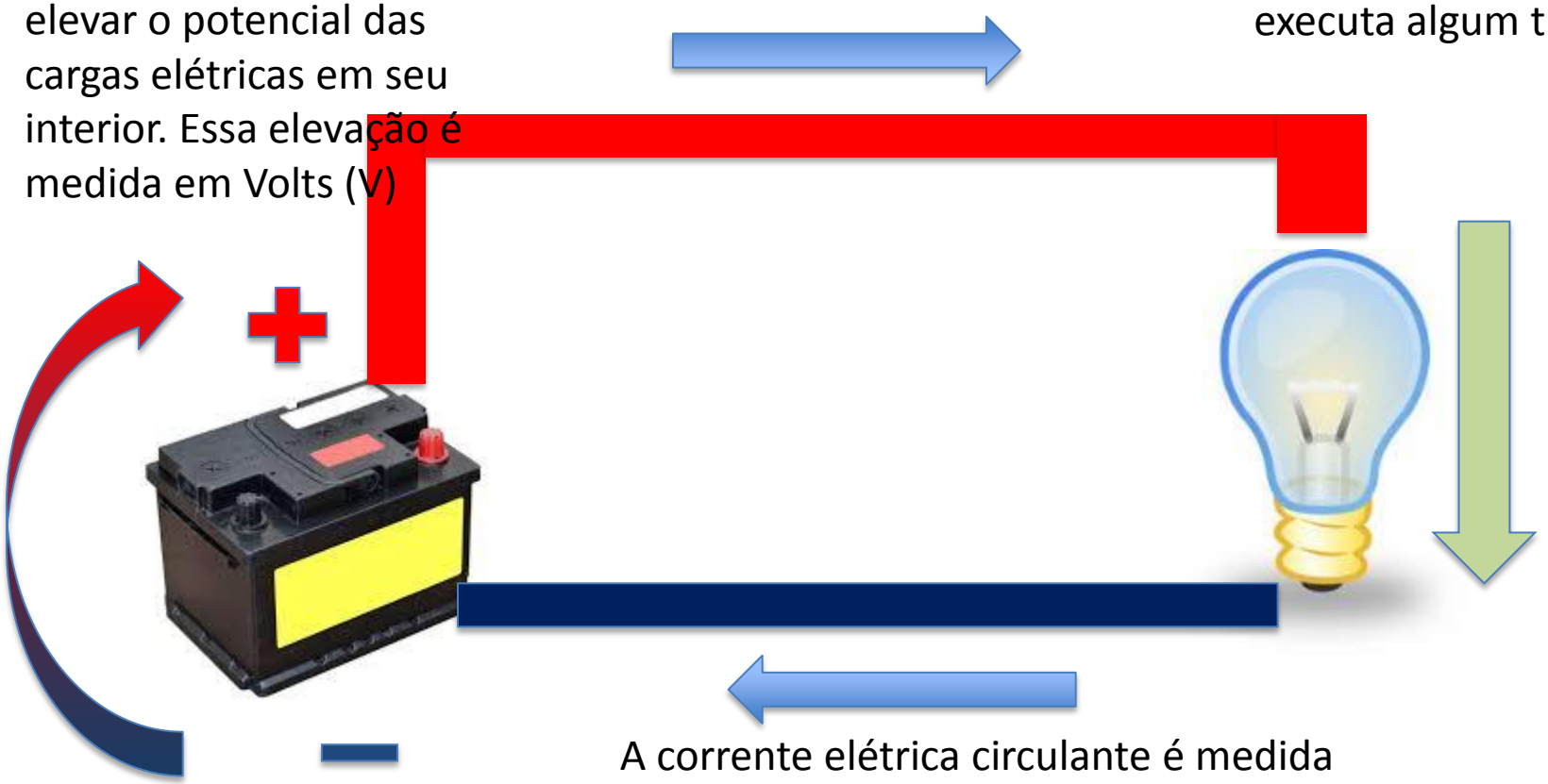


A vazão de água é medida em litros por segundo (l/s)

Circuito elétrico

Uma bateria elétrica usa algum tipo de energia para elevar o potencial das cargas elétricas em seu interior. Essa elevação é medida em Volts (V)

Uma **carga elétrica** usa a diferença de potencial fornecida para gerar uma corrente elétrica que executa algum trabalho



A corrente elétrica circulante é medida em ampères (A), ou em coulombs por segundo (C/s), onde C é a unidade de medida da carga elétrica

Sensores analógico

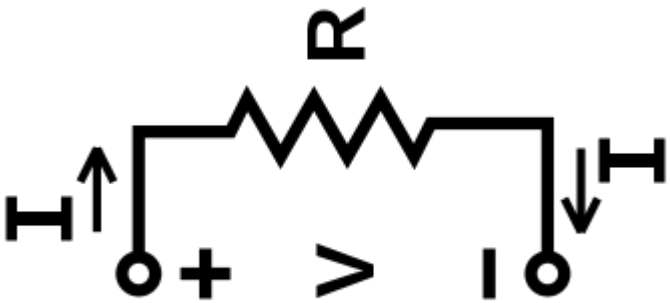
- Os sensores analógicos ou **transdutores elétricos** conseguem capturar um fenômeno físico e transformá-lo em um sinal elétrico (tensão ou corrente) que varia de forma análoga a esse fenômeno.
  - Assim, um microfone é um transdutor eletroacústico, pois transforma a variação da pressão atmosférica em uma corrente elétrica que varia da mesma forma
- Em geral esses sensores não geram sua própria diferença de potencial, por isso dependem do fornecimento externo de tensão para gerar uma corrente que varia de acordo com o fenômeno sendo medido
- Para capturar a medida do sensor, os circuitos digitais como o Arduíno precisam medir o valor dessa corrente elétrica

Medindo tensão e corrente elétrica

- A medida da tensão e da corrente elétrica estão intimamente relacionadas
- Circuitos digitais são muito bons em medir a tensão elétrica, embora, internamente, essa tensão seja transformada numa pequeníssima corrente elétrica.
- Os circuitos digitais que medem a tensão elétrica são chamados de circuitos conversores analógico-digitais (AD), pois convertem um valor de tensão elétrica em um número no formato digital (0s e 1s), composto por uma certa quantidade de bits
- Os conversores AD geram números que variam de 0 a $2^n - 1$, onde n é o número de bits da representação
  - 0 é usado para medir 0V, ou um nível muito pequeno de tensão
  - O valor máximo é usado para medir tensões acima da tensão de referência do conversor
  - No caso do Arduíno, que usa um conversor de 10 bits e tensão de referência de 5V, os valores digitais medidos pelo conversor AD ficarão entre 0 (pata 0V) e 1023 (para 5V)
- Se quisermos medir a corrente elétrica em vez da tensão, precisamos "transformá-la" em tensão elétrica através do uso da Lei de Ohm.

Lei de Ohm

- Um dos dispositivos elétricos mais simples é aquele que usa a corrente elétrica para gerar calor, os chamados **resistores elétricos**.
- Ao receberem uma diferença de potencial em seus terminais, as resistências elétricas permitem a passagem de um valor específico de corrente elétrica.
- À relação entre o valor da tensão elétrica aplicada e da corrente resultante chamamos de resistência elétrica, medida em **ohms** (Ω)
- Assim: $I = V/R$, onde
  - I é a corrente elétrica
  - V é a diferença de potencial aplicada
  - R é o valor da resistência elétrica



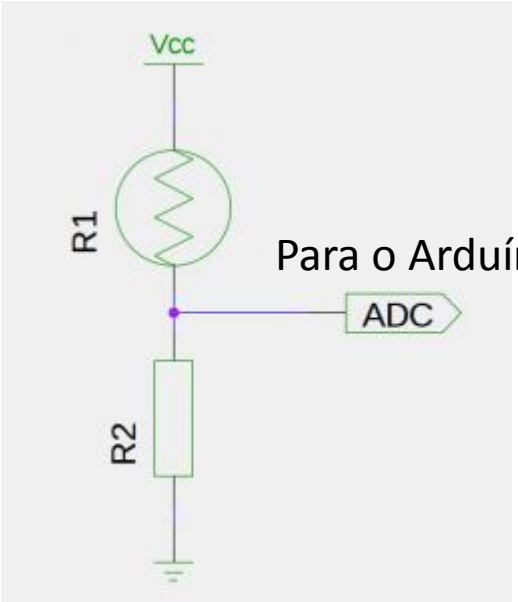
Sensor de luminosidade (LDR)

- Nosso primeiro sensor analógico será o LDR (Light Dependent Resistor), um sensor de luminosidade ambiente
- Ele possui dois contatos elétricos planos separados por uma trilha de sulfito de cádmio.
  - Quando há incidência de luz, essa substância permite a condução de corrente, porém apresentando uma certa resistência elétrica.
  - Quanto maior essa incidência de luz, menor será essa resistência, e portanto, caso haja uma tensão aplicada, maior será a corrente elétrica ali passando.

- O LDR funciona então como um resistor cuja resistência varia de acordo com a incidência de luz.
  - Para medir a luz, precisamos medir essa resistência
  - Para tanto, aplicamos uma diferença de potencial e medimos a corrente que passa

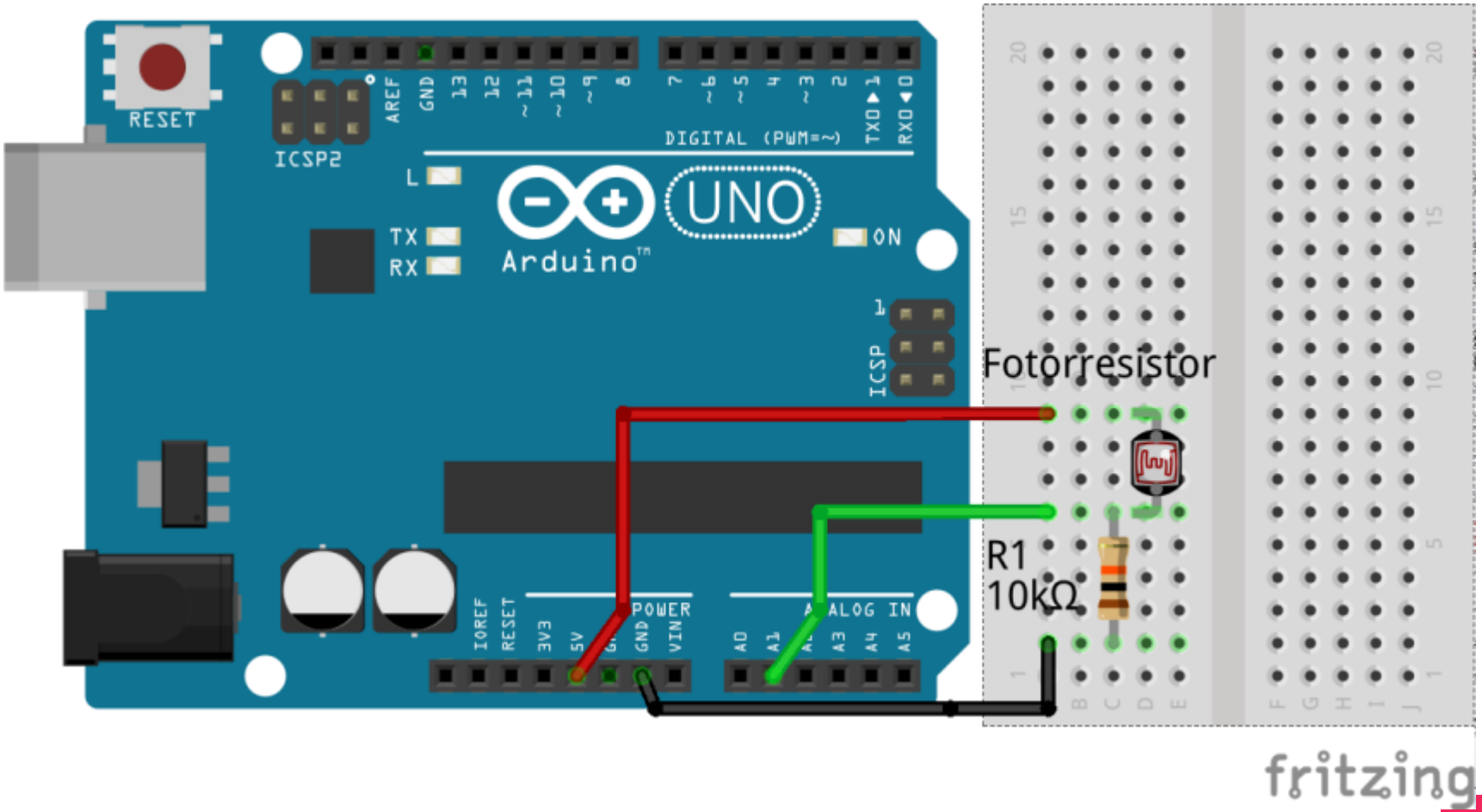
Medindo a luminosidade com o LDR

- Usamos um resistor (R2) ligado em série com o LDR (R1), aplicando uma tensão constante nas extremidades livres
- A resistência equivalente dos dois componentes é a soma das resistências de ambos, e a corrente que passa nos dois componentes é a mesma
- Como a resistência do resistor é constante, a tensão elétrica entre as suas extremidades será proporcional à corrente, que por sua vez é proporcional à luminosidade ambiente

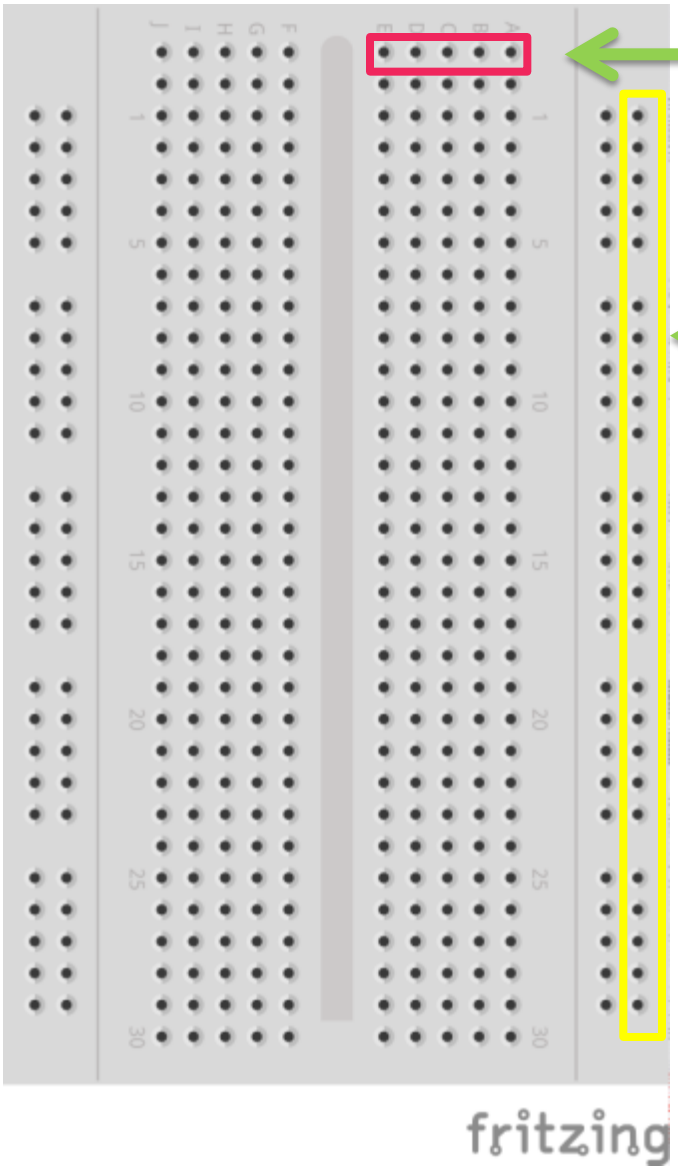


SENSORES ANALÓGICOS

Lendo um sensor de luminosidade



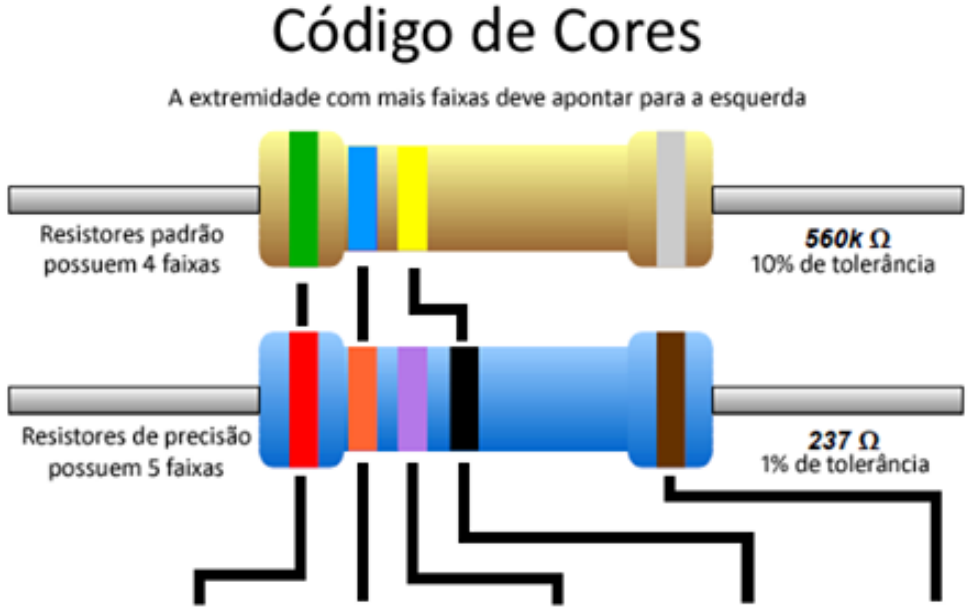
PROTOBOARD



- Faz a conexão elétrica entre componentes
- Cada fileira de 5 orifícios na coluna central forma uma trilha conectada, e isolada das demais trilhas
- Cada coluna de orifícios nas trilhas laterais forma uma trilha de alimentação, e está toda conectada.
  - Geralmente elas são ligadas à tensão de alimentação (5V) ou ao terra (GND ou 0V)
- Dois terminais de componentes plugados à mesma trilha estão eletricamente conectados
  - Nunca podemos conectar dois terminais de um mesmo dispositivo na mesma trilha, pois assim eles estarão em curto-circuito!

Lendo o valor de resistores pelo código de cores

- Verificar através da tabela de cores o algarismo correspondente à cor do primeiro anel, que será o primeiro dígito do valor da resistência.
- Verificar através da tabela de cores o algarismo correspondente à cor do segundo anel, que será o segundo dígito do valor da resistência.
- Determinar o valor para multiplicar o número formado pelos itens 1 e 2 pela cor do anel multiplicador (3ª ou 4ª faixa)
- Verificar a porcentagem de tolerância do valor nominal da resistência pela cor do último anel.
- Ex: cores marrom (1), preto (0), laranja (x1k) e dourado: 10k ohms de resistência com tolerância de 5%



Cor	1ª Faixa	2ª Faixa	3ª Faixa	Multiplicador	Tolerância
Preto	0	0	0	x 1 Ω	
Marrom	1	1	1	x 10 Ω	+/- 1%
Vermelho	2	2	2	x 100 Ω	+/- 2%
Laranja	3	3	3	x 1K Ω	
Amarelo	4	4	4	x 10K Ω	
Verde	5	5	5	x 100K Ω	+/- .5%
Azul	6	6	6	x 1M Ω	+/- .25%
Violeta	7	7	7	x 10M Ω	+/- .1%
Cinza	8	8	8		+/- .05%
Branco	9	9	9		
Dourado				x .1 Ω	+/- 5%
Prateado				x .01 Ω	+/- 10%

Fonte: http://eletronsdadepressao.blogspot.com.br/2015/01/codigo-de-cores-de-resistores.html

SENSORES ANALÓGICOS

Lendo um sensor de luminosidade

```
int sensor = A1;//Pino analógico em que o sensor está conectado

void setup(){
  Serial.begin(9600);
}

void loop(){
  //Lendo o valor do sensor.
  int valorSensor = analogRead(sensor);

  //Exibindo o valor do sensor no serial monitor.
  Serial.println(valorSensor);

  delay(500);
}
```

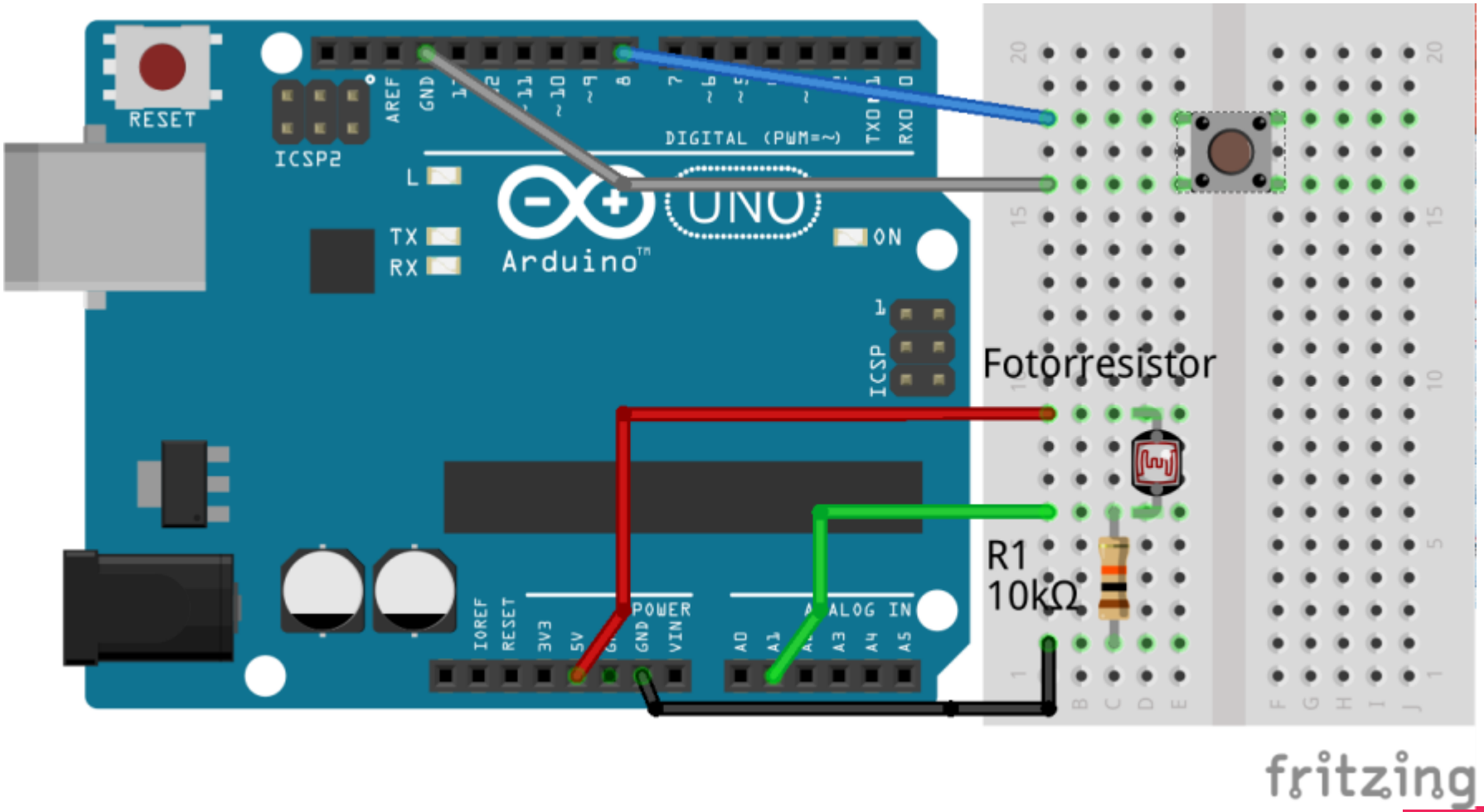
SENSORES DIGITAIS

Fazendo leitura de um botão

- Ligue um terminal do botão na porta 8, e a outra no GND (ver figura logo mais)
- Execute o programa a seguir e acompanhe pelo monitor serial:
  - Experimente apertar o botão e ver a saída

```
int inPin = 8; //entrada digital na porta 8
int val = 0;
void setup() {
  Serial.begin(9600);
  pinMode(inPin, INPUT_PULLUP); //porta 8 vira entrada
}
void loop(){
  val = digitalRead(inPin);   // read the input pin
  Serial.println(val);
  delay(2000);
}
```

SENSORES DIGITAIS



Saídas digitais

Fazem o acionamento de dispositivos do tipo liga/desliga

- As saídas D0 a D13 permitem a leitura ou escrita de valores lógicos. Por isso são chamadas de GPIO (General-Purpose Input and Output)
- As portas D0 (RX0) e D1 (TX0) são destinadas para a comunicação serial – via cabo USB ou comunicação com shields, por exemplo
- São configuradas como saídas digitais na função setup() através da chamada:
  - `pinMode(numero, OUTPUT);`
- Podem acionar dispositivos que necessitem de uma informação digital (0 ou 1) para seu acionamento, por exemplo:
  - Luz LED
  - Relê para acionamento de equipamentos elétricos
  - Dispositivo de acionamento de motores (drives)
- Na saída porta D13 há um LED conectado, que acende assim que a porta esta acionada
- Vamos executar o exemplo: `File` → `Examples` → `01.Basic` → `Blink`
  - `digitalWrite(HIGH | LOW);` aciona ou desliga a saída digital

Exemplo: piscar o LED da porta D13

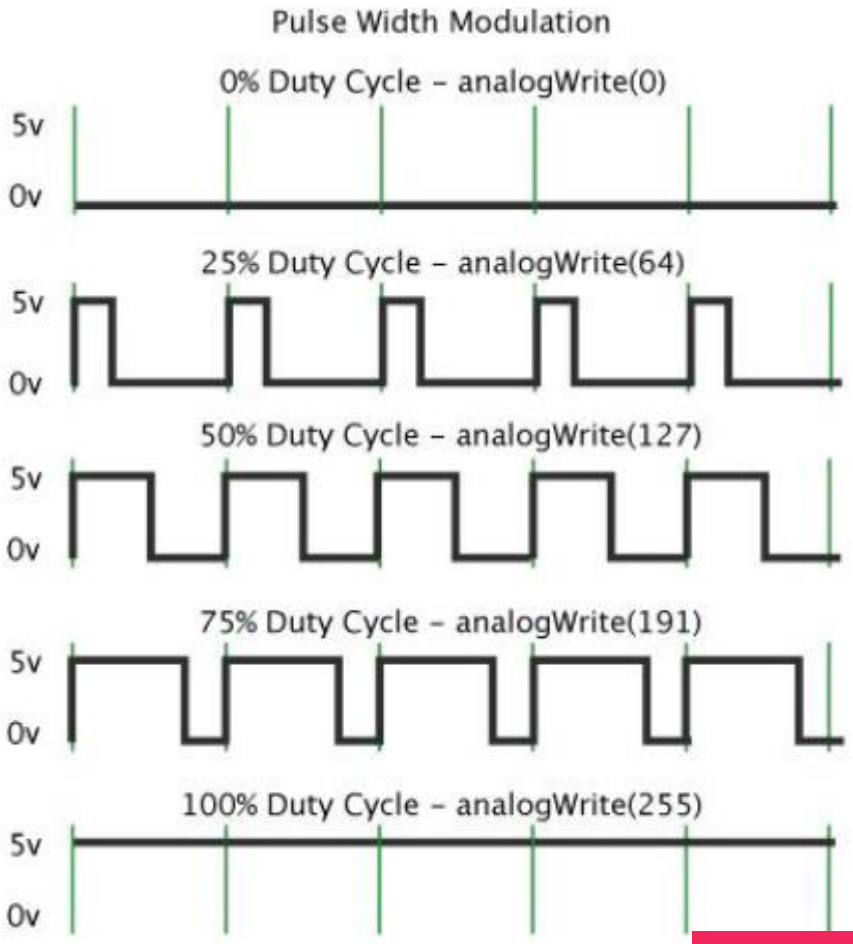
```
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}
// the loop routine runs over and over again forever:
void loop() {
  // turn the LED on (HIGH is the voltage level)
  digitalWrite(led, HIGH);
  delay(1000);                    // wait for a second
  // turn the LED off by making the voltage LOW
  digitalWrite(led, LOW);
  delay(1000);                    // wait for a second
}
```

SAÍDAS ANALÓGICAS (PWM)

Fazem o acionamento de dispositivos que possam ser controlados por PWM

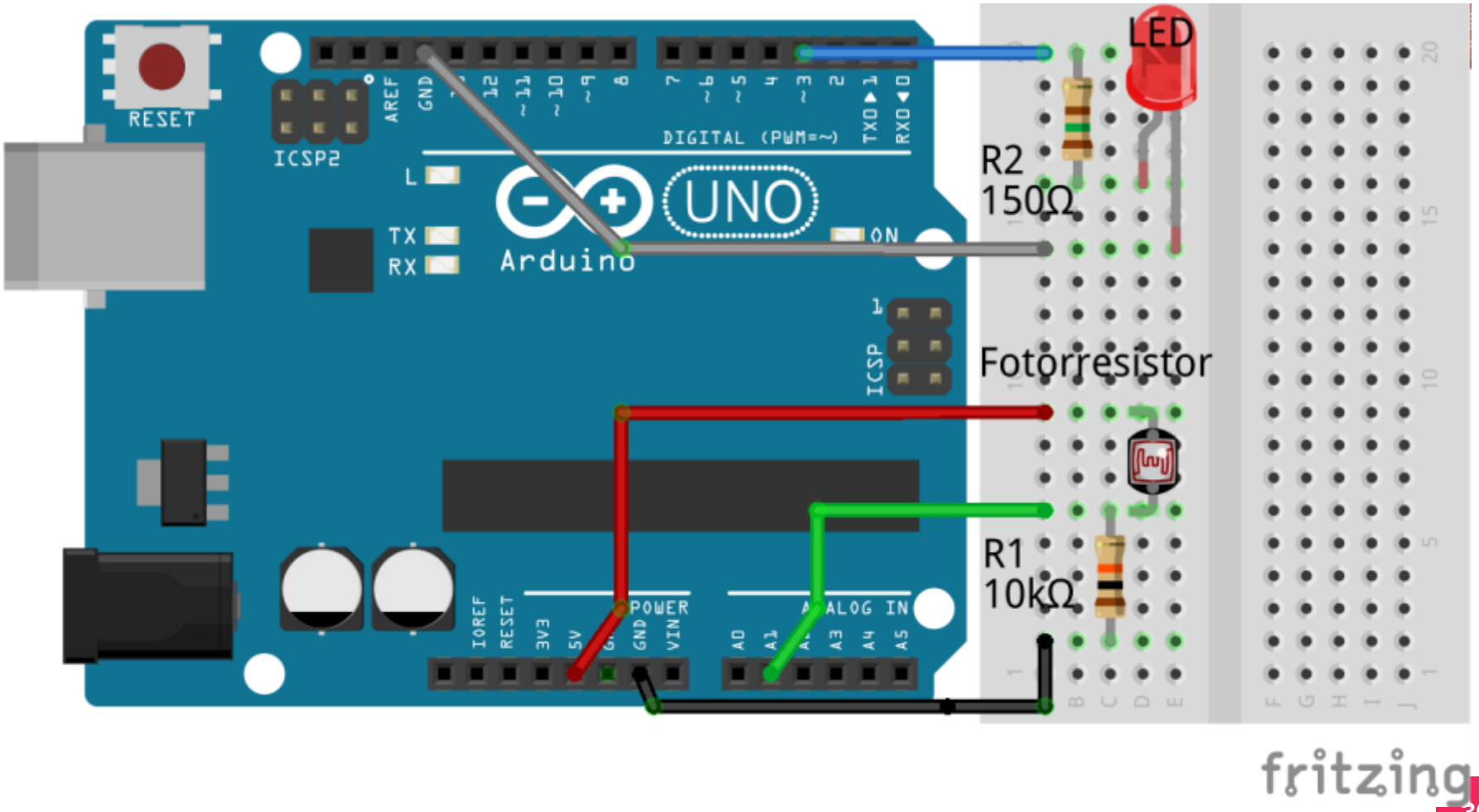
- As portas digitais (GPIO) marcadas com um ~ (til) possuem a capacidade de servirem de saída do tipo PWM – Pulse-Width Modulation
- Uma saída PWM automaticamente alterna períodos em que ela esta ligada (HIGH) e em que está desligada (LOW). A proporção do tempo em que a saída está ligada em relação ao tempo total do ciclo é chamad de Duty Cycle, e varia de 0 a 100%
- Exemplos: motor de passo, iluminação LED ou acionador dimerizável de lâmpada, LED RGB (seleciona a cor do LED)
- Para escrever um valor de 0 a 255 na saída PWM, esta deve estar configurada como uma saída digital, sendo que a escrita se dá da forma:

```
analogWrite(numero, valor);
```



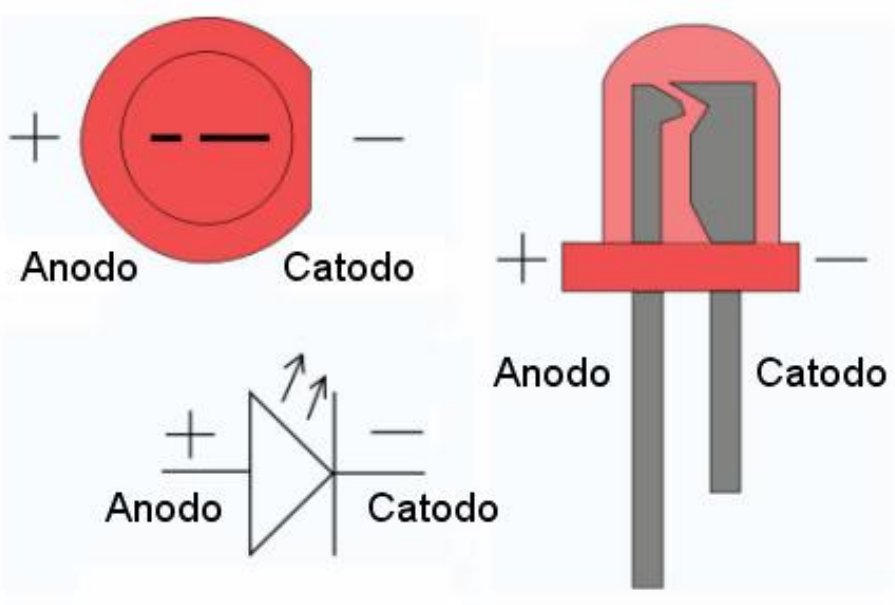
SAÍDA ANALÓGICA

Lendo um sensor de luminosidade



Instalação do LED

- Deve-se sempre interpor um resistor de 100 a 150 ohms em série ao LED, para evitar uma sobretensão no mesmo
  - O resistor pode estar interposto tanto ao catodo quanto ao anodo, indistintamente
- O LED possui polaridade. Seus terminais são:
  - Anodo: correspondente ao polo positivo, deve ser ligado à porta controladora
  - Catodo: correspondente ao polo negativo, deve ser ligado ao GND (0V)



SAÍDA ANALÓGICA

Controlando a intensidade do LED a partir do sensor de luminosidade

```
int sensor = 1;//Pino analógico em que o sensor está conectado
int led = 3; //Pino em que o led está conectado

void setup(){
  Serial.begin(9600);
  pinMode(led,OUTPUT);
}
void loop(){
  //Lendo o valor do sensor.
  int valorSensor = analogRead(sensor);
  //Exibindo o valor do sensor no serial monitor.
  Serial.println(valorSensor);
  //Acionando o LED: quanto menos luz externa, mais forte o LED
  //map(valor,deEscalaAnt,ateEscalaAnt,deNovaEscala,ateNovaEscal)
  analogWrite(led, map(valorSensor,0,1023,255,0));
  delay(50);
}
```

REFERÊNCIAS

- http://www.telecom.uff.br/pet/petws/downloads/tutoriais/arduino/Tut_Arduino.pdf
- http://arduino.cc/en/Reference/HomePage
- http://eletronsdadepressao.blogspot.com.br/2015/01/codigo-de-cores-de-resistores.html

FIAP