

Manual de

Multimédia Tangível I

João Cordeiro

Título | Manual de Multimédia Tangível I

Autor | João Cordeiro

Contacto | joao.cordeiro@uevora.pt

Data da primeira edição | Setembro 2024

Escola de Artes
Universidade de Évora

Índice

Sobre este manual.....	1
C1. Introdução à unidade curricular Multimédia Tangível I	2
C2. Conceitos operatórios	14
C3. Introdução à eletricidade	20
C4. Introdução à eletrónica: circuitos elétricos em ambiente virtual	26
C5. Eletrónica analógica: circuitos eletrónicos com componentes físicos.....	31
C6. Eletrónica digital: introdução aos microcontroladores.....	36
C7. Eletrónica digital: atuadores.....	46
C8. Eletrónica digital: sensores	64
C9. Controlo de software com um microcontrolador	75

Sobre este manual

O presente manual foi concebido para auxiliar o estudante da unidade curricular de Multimédia Tangível I a acompanhar os conteúdos lecionados e consequentemente adquirir as competências previstas na ficha de unidade curricular.

Não é objetivo deste texto ser um manual exaustivo sobre o tema da multimédia tangível, pretende-se antes que este manual forneça uma base anotada dos fundamentos desta área disciplinar aplicada ao contexto artístico, incluindo exemplos de obras e os princípios básicos da eletrónica e programação informática.

O manual encontra-se estruturado em capítulos que se podem considerar sequenciais e como correspondendo – ainda que de forma maleável – às sessões lecionadas ao longo de um semestre.

Esta versão do texto nada mais é que um rascunho, um esboço incompleto de um edifício que demorará a ficar completo mas que aqui se pretende delinear em sólido alicerce.

C1. Introdução à unidade curricular Multimédia Tangível I

Sobre o capítulo: conteúdos, objetivos e recursos

Conteúdo

- Apresentação do docente
- Apresentação da UC
- Obras de referência da arte interativa

Objetivos de aprendizagem

- Conhecer o âmbito disciplinar técnico e artístico da UC;
- Conhecer o modelo de avaliação da UC;
- Conhecer o modelo de ensino-aprendizagem a seguir no decorrer da aula;
- Conhecer algumas obras de referência no domínio da arte interativa;

Recursos

- Slideshow sobre apresentação do professor
- Google form para recolher informação sobre os alunos
- FUC da UC
- Sebenta da UC (com programa)
- Google Classroom da UC
- Moodle da UC
- Enunciados das avaliações

Sobre o docente

João Cordeiro é doutorado em Ciência e Tecnologia das Artes pela Escola das Artes da Universidade Católica Portuguesa. Ao longo da sua carreira artística e académica tem desenvolvido projetos na área da arte multimédia, que se desdobram pelas disciplinas do som, vídeo, animação e arte interativa.

Âmbito da UC

A UC Multimédia Tangível I insere-se no âmbito do conhecimento científico sobre os modelos de utilização e interação com tecnologia multimédia baseados em dispositivos tangíveis, isto é, modelos de interação em que as interfaces privilegiam a entrada e saída de dados através de transformações físicas, quer no sentido de se adaptarem à ergonomia do utilizador quer se baseiem em alterações morfológicas correspondentes à informação de saída.

No final desta unidade curricular, os alunos deverão ter adquirido competências de proficiência intermédia das técnicas e tecnologias utilizadas em projetos de multimédia tangível, ou numa perspetiva mais alargada, processos computacionais e informação digital transferidos para forma física, especificamente, na vertente das artes, ciência e tecnologia, com vista à criação de instalações multimédia performáticas e espaços reativos.

Programa

O programa da unidade curricular Multimédia Tangível I corresponde – *grosso modo* – ao índice deste manual, nomeadamente:

	Plano de sessão
1	Apresentação da UC
2	Introdução aos conceitos operatórios
3	<i>Reinscrição às UCs.</i> Introdução à eletricidade

4	Introdução à eletrónica analógica (virtual)
5	Introdução à eletrónica analógica (física)
6	Introdução à eletrónica digital: microcontroladores
7	Introdução a sensores analógicos em microcontroladores
8	Introdução a sensores e atuadores digitais
9	Avaliação Intercalar 1: teste sobre conhecimentos técnicos Ligação entre microcontroladores e o protocolo MIDI/OSC
10	Ligação entre microcontroladores e Processing / Touch Designer
11	Outros inputs: webcams, controladores de consolas
12	Avaliação intercalar 2: apresentação do conceito de projeto
13	Acompanhamento de projeto
14	Acompanhamento de projeto

Nota: por motivos imprevistos, a calendarização poderá alterar-se. As datas oficiais de avaliação deverão ser consultadas no Moodle.

Avaliação

A avaliação pode adotar um dos seguintes regimes:

Regime de Avaliação Contínua (AC)

- 20% - Participação em aula: postura, assiduidade, pontualidade
- 20% - Avaliação intercalar 1: teste sobre fundamentos de eletrónica
- 20% - Avaliação intercalar 2: proposta de projeto final
- 40% - Projeto final

Regime de Avaliação Final (AF)

- 20% - Avaliação intercalar 1: teste sobre fundamentos de eletrónica (AC)
- 20% - Avaliação intercalar 2: proposta de projeto final (AC)
- 60% - Exame escrito teórico-prático sobre fundamentos histórico-estéticos relacionados com a multimédia tangível e exercícios/desafio práticos de eletrónica.

Na época normal de avaliação os alunos podem optar pelos regimes de Avaliação Contínua ou Avaliação Final. As épocas de recurso, especial e extraordinária guiam-se pelo regime de Avaliação Final.

No regime de Avaliação Final a nota dos elementos avaliação intercalar 1 e avaliação intercalar 2 é obtida através do trabalho desenvolvido e avaliado durante o período de Avaliação Contínua.

Enunciado da Avaliação Intercalar II

Tarefa

Os alunos – individualmente ou em pares - deverão conceptualizar, planear e prototipar a obra interativa/reactiva ou a ferramenta que irão implementar na avaliação Projeto Final. Deverão preparar um dossier de projeto com os seguintes elementos:

- Conceito da obra/ferramenta
- Obras/artistas de referência
- Recursos necessários à sua implementação e apresentação
- Planificação temporal do trabalho a desenvolver (tarefas/semanas)

Critérios de avaliação

- Qualidade da escrita, formatação e formatação do dossiê
- Pertinência e originalidade do conceito, enquadrado no contexto da criação artística contemporânea
- Qualidade/correção do conteúdo: planificação, obras de referência, etc.
- Mérito da apresentação oral/audiovisual

Data e modo de entrega

O dossier deverá ser submetido em PDF, no Google Classroom, antes da 12ª sessão. Na 12ª sessão os alunos deverão apresentar oralmente (e se oportuno com recursos audiovisuais e protótipos) o conteúdo do dossier.

Enunciado do Projeto Final

Tarefa

Os alunos deverão conceptualizar, planear e implementar – individualmente ou em pares - uma obra artística interativa/reactiva ou um instrumento/ferramenta com o qual possam criar uma obra. Neste caso, ambos os elementos (instrumento e obra resultante) formam o objeto de avaliação.

Critérios de avaliação

- Complexidade técnica (demonstrativa de competências)
- Quantidade de trabalho envolvido no projeto
- Enquadramento da proposta no contexto da arte contemporânea
- Qualidade técnica-estética da obra

Data e modo de entrega

Apresentar a obra ao júri durante o período de avaliação final (em data a definir).

Equipamento necessário

Pela especificidade técnica da unidade curricular Multimédia Tangível I, existe um conjunto de equipamentos técnicos, ferramentas e consumíveis que serão importantes para a boa aquisição de conhecimentos e processo de avaliação (Projeto Final).

A Escola de Artes dispõe de uma quantidade significativa de equipamento que muito contribuirá para a dinâmica de ensino-aprendizagem, no entanto esse equipamento poderá não ser suficiente para todos os alunos/projetos e poderá não estar sempre disponível para o desenvolvimento dos projetos de avaliação.

Assim, será desejável que os alunos adquiram os seguintes equipamentos:

- Computador portátil
- Microcontrolador

- Placa de prototipagem (*breadboard*)
- Cabos de ligação
- Fonte de alimentação regulável
- Alicates de descarnar
- Vários componentes eletrónicos (de acordo com o projeto a desenvolver)

Outros equipamentos que são úteis ao trabalho a desenvolver no âmbito da UC:

- Kit de soldar (a estanho)
- Multímetro

Fornecedores

Existem inúmeros fornecedores de materiais eletrónicos. Aqui fica uma pequena lista com fornecedores fidedignos:

- <https://mauser.pt/>
- <https://www.ptrobotics.com/410-outlet>
- <https://pt.mouser.com/>
- <https://www.adafruit.com/>

Exemplos de obras e instrumentos com elementos tangíveis

Neste subcapítulo são apresentados exemplos de obras de arte e ferramentas/instrumentos com características que podemos definir como tangíveis, isto é, que utilizam elementos físicos (normalmente por parte do visitante) que alteram e fazem acontecer a obra. Na maioria dos exemplos apresentados, encontramos-nos no domínio da arte interativa, que é uma forma de expressão artística que beneficia enormemente das tecnologias da multimédia tangível.

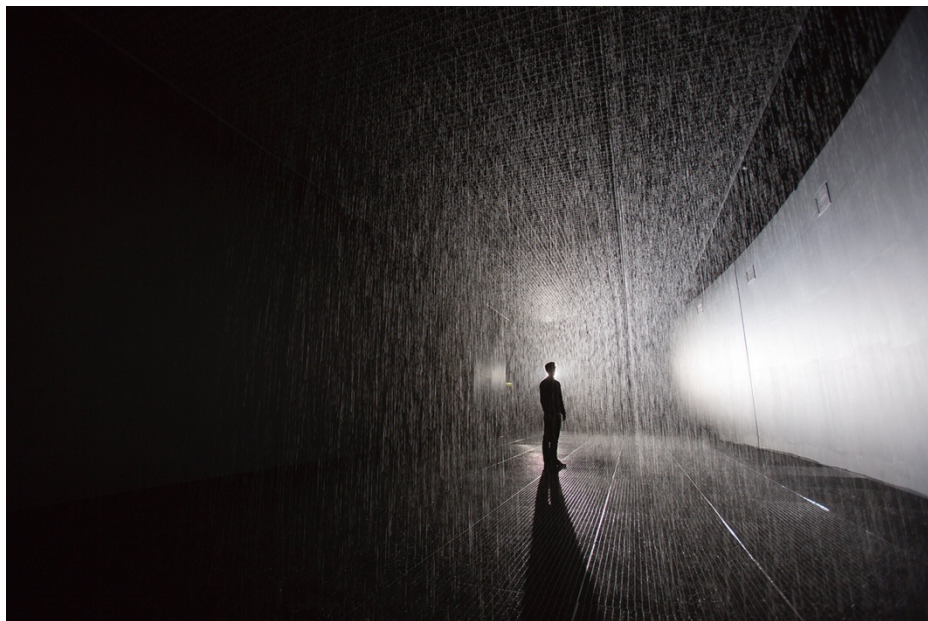


Figura 1 - Rain Room, 2012, por Random International.

Rain Room é uma obra de arte interativa criada em 2012 pelo coletivo Random International. Consiste numa sala escura de dimensão considerável, com pouca luz, em que vários expressores no teto simulam a queda de chuva. Ao entrar na área dos expressores, os que se encontram por cima do visitante desligam-se, fazendo com que este não se molhe.

- <https://www.random-international.com/rain-room-2012>
- <https://vimeo.com/51830893>



Figura 2 - Text Rain, 1999, por Romy Achituv & Camille Utterback

Text Rain é uma obra de videoarte interativa criada por Romy Achituv e Camille Utterback, em 1999. Consiste num ecrã situado em frente ao visitante onde é projetada a imagem deste com letras que lhe caem em cima, as quais alteram a sua queda com base na silhueta da pessoa na imagem. As letras formam frases de um poema sobre o corpo e a linguagem.

- <https://camilleutterback.com/projects/text-rain/>
- https://www.youtube.com/watch?v=f_u3sSffS78



Figura 3 - Lines, 2016, Anders Lind

Lines é uma obra de arte sonora interativa criada por Anders Lind, em 2016. Consiste num painel/parede com linhas horizontais sobre as quais os visitantes podem colocar objetos (pontos), os quais vão alterar o som que a peça produz.

- <https://www.soundlikelind.se/sound-artist>
- <https://www.youtube.com/watch?v=hP36xoPXDnM>



Figura 4 - Wooden Mirror, 1999, por Daniel Rozin

Wooden Mirror é uma obra de arte interativa criada por Daniel Rozin em 1999. Consiste num conjunto de blocos de madeira que se assemelham a pixéis e que se adaptam à imagem que está em frente (do visitante), como se fosse um espelho.

- <https://www.smoothware.com/danny/woodenmirror.html>
- https://www.youtube.com/watch?v=0o_9CHYeRvI

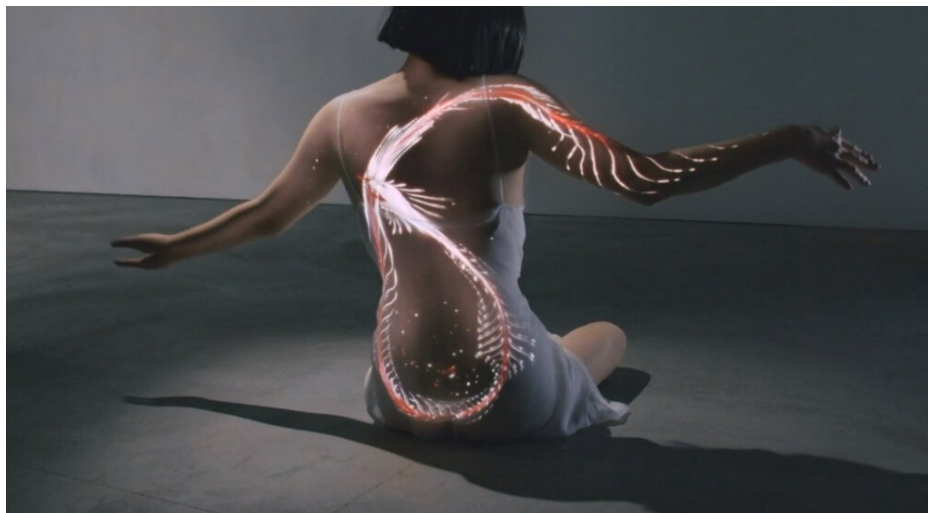


Figura 5 - *Parted*, 2020, por Daisuke Hashimoto, Paul Lacroix, Yae Doi e Mari Katayama

Parted é uma obra intermedia que combina *video mapping* adaptativo, vídeo arte, performance e som. Foi desenvolvida por Daisuke Hashimoto, Paul Lacroix, Yae Doi e Mari Katayama em 2020. A projeção de vídeo no corpo da performer adapta-se ao seu movimento.

- <http://parted.tokyo/>
- <https://www.youtube.com/watch?v=BndWx6UUtMQ>

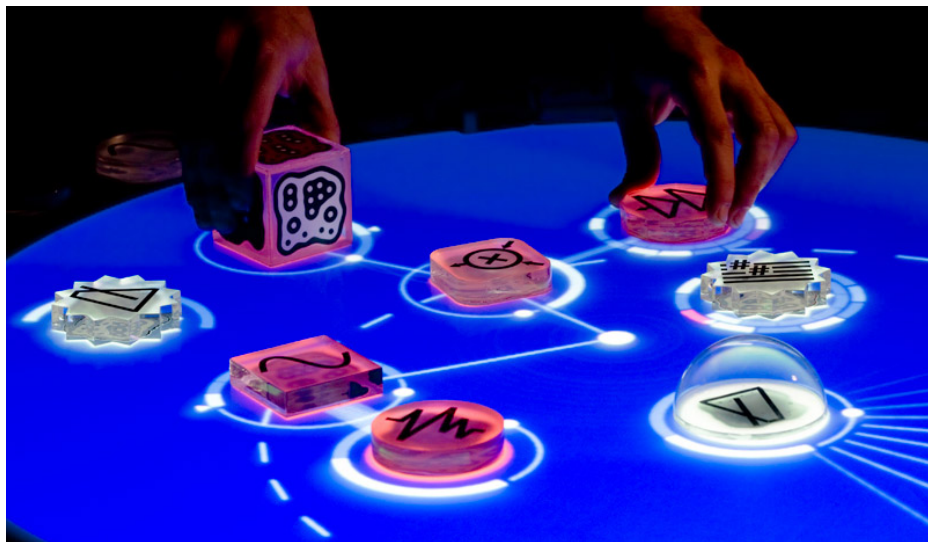


Figura 6 - Reactable, 2005, Sergi Jordà, Martin Kaltenbrunner, Günter Geiger e Marcos Alonso

Reactable é um instrumento musical digital apresentado pela primeira vez em 2005, criado por Sergi Jordà, Martin Kaltenbrunner, Günter Geiger e Marcos Alonso. Diferentes peças são colocadas em cima de um ecrã despoletando cada uma delas um evento sonoro, os quais podem ser samples, sons sintetizados ou processamento. É utilizado em performances musicais.

- <https://reactable.com/>
- https://www.youtube.com/watch?v=ptyH_RX5Jl0

C2. Conceitos operatórios

Sobre o capítulo

Conteúdo

- Glossário de termos relacionados
- Teoria de processamento de sinal aplicada às obras de arte
- Apresentação das principais tecnologias de input, output e processamento.

Objetivos de aprendizagem

- Adquirir noções sobre os principais conceitos relacionados com multimédia tangível.
- Aplicar a teoria de processamento de sinal à análise de obras interativas.
- Identificar e distinguir tecnologias de input, output e processamento.

Glossário de conceitos

As definições apresentadas a seguir pretendem dotar o aluno das noções fundamentais que lhe permitam um entendimento operativo dos respetivos conceitos. Não se pretende aqui optar por uma ou outra definição a partir de referências bibliográficas, pois cada um dos conceitos apresentados se prestaria a longas e complexas discussões epistemológicas que iriam muito para além dos objetivos da unidade curricular.

Conceitos técnicos

Multimédia: conjugação de meios audiovisuais (imagem, som, texto, luz, etc.) com um sistema computacional que geralmente lhes confere modos de apresentação não-lineares e/ou interativos.

Interface: a tecnologia que permite e define o tipo de interação entre o homem e a máquina. É geralmente distinguido entre interfaces de entrada (do homem para a máquina) e interfaces de saída (da máquina para o homem).

Interface tangível: diz-se do interface entre homem e computador que incorpora - como parte do seu paradigma de interação - elementos físicos que permitem manipular ou apresentar informação digital.

Visão por computador: o conjunto de técnicas e tecnologias que permite ao computador, através de uma câmara ou outros dispositivos de entrada, identificar e categorizar os diferentes elementos e atividades presentes numa imagem.

Audição por computador: o conjunto de técnicas e tecnologias que permite ao computador, através de microfones ou outros dispositivos de entrada, identificar e categorizar os elementos presentes num determinado ambiente sonoro.

Sistema reativo: diz-se de um sistema concebido para responder a estímulos criados por um utilizador ou outra entidade.

Sistema interativo: diz-se de um sistema concebido para responder a estímulos provocados por um utilizador (ou outra entidade), alterando-se e evoluindo em função dos estímulos, convocando o utilizador a novas interações.

Sistema imersivo: diz-se de um sistema concebido para imergir sensorialmente o utilizador num universo com características multissensoriais (som, imagem, háptico, temperatura, etc.)

Conceitos artísticos

Arte interativa: diz-se da forma de expressão artística que convoca o visitante a modificar a obra de arte para, de certa forma, a completar ou fazer acontecer. É geralmente associada a arte computacional, mas podem considerar-se obras interativas sem necessidade da presença de um computador.

Arte generativa: diz-se da forma de expressão artística cujas obras têm na sua génese uma formulação algorítmica que permite o desenvolvimento semiautónomo da obra, sem ou com reduzida interferência do autor.

Circuit bending: nome dado à manipulação de circuitos eletrónicos contruídos para um determinado fim com vista à alteração das suas funcionalidades, geralmente dotando-os de capacidades de geração/manipulação de sinais áudio e/ou vídeo.

Glitch art: nome dado à expressão estética que procura o erro como forma de belo, quer através de utilização de sistemas que introduzam erros e distorções ou através da criação de obras que emulem sistemas que produzam erros.

Teoria de DSP aplicada ao estudo de obras artísticas interativas

O processamento de sinal é uma disciplina que se debruça sobre o estudo de sinais (normalmente elétricos); no âmbito digital, utiliza-se o termo DSP (*digital signal processing*).

A base do DSP assenta na seguinte ideia: um sinal A (entrada/input) é manipulado pelo processador P de onde resulta o sinal A' (saída/output). (Figura 7, p.16)



Figura 7 - Diagrama de DSP

O estudo de DSP no contexto desta unidade curricular é relevante por dois motivos: ajuda-nos entender como os sinais elétricos funcionam num circuito eletrónico e oferece-nos um bom quadro de referência para analisar obras interativas, assumindo o estímulo do visitante como um sinal de entrada e a resposta da obra como o sinal de saída¹.

Exemplo prático

Analisemos a obra *Wooden Mirror* de Daniel Rozin (p.11):

O sinal de entrada (A) é composto pelo estímulo causado pela presença do visitante em frente à obra. Trata-se de um exemplo de visão por computador (ver Glossário de conceitos, p.14) onde a imagem do visitante é captada por uma câmara posicionada centralmente na peça.

O processamento (P) desse sinal baseia-se na tradução dos pixéis captados pela câmara em movimentos mecânicos que fazem girar as placas de madeira.

¹ Atenção, a utilização da teoria de DSP neste contexto tem valor de analogia, pois do ponto de vista matemático não seria totalmente correto entender a ação de uma pessoa sobre um interface como uma transferência de sinal.

O sinal de saída (A') corresponde à resposta da obra ao estímulo A, ou seja, a recriação da imagem do visitante através de pixéis de madeira.

Exercício

Em grupos de ± 3 alunos, analisar à luz da teoria de DSP as obras/instrumentos referidos na secção *Exemplos de obras e instrumentos com elementos tangíveis* (p.7), identificando as principais tecnologias e técnicas adotadas em cada um dos momento (A, P, A'), assim como as suas implicações estéticas e comunicacionais (nunca esquecer esta parte!).

Dispositivos de entrada, saída e estratégias de processamento

Na secção anterior ficou demonstrado que a teoria do DSP pode ser útil para nos ajudar a desconstruir obras de arte interativa, analisando cada um dos elementos de entrada, processamento e saída, sem nunca esquecer as suas implicações estéticas.

Nesta secção vamos sistematizar um pouco mais as estratégias e tecnologias que participam em cada momento.

Dispositivos de entrada (input)

Os dispositivos de entrada são todos aqueles que captam variações no mundo físico/sensível e que permitem a transferência dessas variações para o universo dos sinais (analógicos e digitais).

Um interruptor elétrico, por exemplo, permite converter a pressão exercida pelo dedo do visitante num evento de saída como seja o acender de uma lâmpada. Um termostato permite utilizar a variação de temperatura de uma casa para ligar/desligar um aquecedor.

Os dispositivos de entrada encapsulam uma panóplia de sensores e estratégias para detetar variações no mundo físico/sensível. Por exemplo:

- **Pressão** – botões (interruptores, controladores de jogos, botões de rato...), teclados, barómetros (pressão atmosférica), etc.

- **Movimento** – sensores de infravermelho, câmaras de vídeo, giroscópicos, sensores de rotação;
- **Temperatura** – câmaras térmicas, sensores de temperatura;
- ...

Dispositivos de saída (output)

Os dispositivos de saída são normalmente os que definem a tipologia da obra. Por exemplo, se o dispositivo de saída debita som diz-se que se trata de uma de arte sonora interativa (*interactive sound art*).

Podem ser usado inúmeros dispositivos de saída nas obras de arte interativa, sendo usual combinar mais do que um dispositivo/estratégia na mesma obra. Por exemplo:

- **Dispositivos visuais** – ecrãs, projetores de vídeo, luzes;
- **Dispositivos mecânicos** – motores, atuadores;
- **Dispositivos sonoros** – colunas;
- ...

Alguns dispositivos são mistos, ou seja, permitem captar variações no mundo sensível e disponibilizar informação, como é o caso dos ecrãs de toque que equipam os nossos telemóveis, ou o interruptor elétrico do sótão que emite uma pequena luz para nos indicar que a luz do sótão está acesa. Neste caso específico, essa pequena luz tem valor de *feedback*.

Processamento

O conceito de processamento deverá aqui ser entendido à luz da analogia com DSP e não no sentido estrito do termo associado às ciências da computação, isto porque existem obras onde a componente da interatividade poderá não passar sequer por um dispositivo computacional (veja-se o exemplo do interruptor elétrico).

No entanto, será útil elencar – de forma abstrata - alguns dos processos/estratégias que podem ser implementadas numa obra interativa:

- **Geração** – o estímulo de entrada dá aso à geração de elementos de saída (imagem, som, movimento);

- **Manipulação** – os estímulos de entrada são manipulados/transformados e direcionados para a saída (por exemplo, a imagem capturada por uma câmara é invertida e projetada);
- **Análise** – a principal função do processamento é analisar os estímulos de entrada (por exemplo, um algoritmo de visão por computador que identifique o número de pessoas na imagem captada por uma câmara):
- ...

Exercício

Em modo de *brainstorming*, toda a turma contribui com ideias de dispositivos de entrada, saída e estratégias de processamento. A informação compilada é organizada num mapa mental.

C3. Introdução à eletricidade

Sobre o capítulo

Conteúdo

- Princípios básicos de eletrônica
- Sistema eletrônicos em ambiente digital

Objetivos de aprendizagem

- Conhecer os princípios básicos de eletrônica
- Construir circuitos básicos de eletrônica em ambiente virtual

Recursos

- Tinkercad (software)

Princípios de eletricidade

Uma definição de eletricidade

A eletricidade é um tipo de energia que consiste no movimento de elétrons entre dois pontos em que se observa uma diferença de potencial, proporcionando uma corrente elétrica.

AC vs. DC

Um dos primeiros conceitos que é importante entender é a diferença entre corrente alternada (AC) e corrente contínua (DC). A corrente alternada é o tipo de corrente que existe nas tomadas de casa. É um tipo de corrente elétrica que alterna constantemente de direção, sendo por isso mais fácil, barata e segura a

sua distribuição ao longo de largas distâncias, embora possa criar interferências em equipamentos mais sensíveis. A corrente contínua é geralmente utilizada em situações em que baixas voltagens são requeridas, sendo por isso o tipo de eletricidade encontrada nos transformadores e pilhas que alimentam a maioria dos circuitos eletrônicos. Nestes casos é potencialmente mais estável, constante e livre de interferências que a corrente alternada. É também mais fácil de armazenar (daí ser usada em baterias), e em baixos valores de tensão e corrente não constitui um perigo para a saúde.

É possível converter AC em DC e vice-versa.

Unidades do Sistema Internacional: Volt, Ampere, Watt e Ohm

O volt e o ampere são duas unidades fundamentais para compreender e trabalhar com eletricidade. É importante perceber o que elas representam e como se interligam antes de começar a trabalhar com circuitos, pois só assim vamos evitar queimar os nossos componentes!

Como referido anteriormente, a eletricidade é um fluxo ordenado de eletrões. Para nos ajudar a compreender melhor a natureza deste fluxo é comum socorreremo-nos da analogia da água numa mangueira. Assim, se o fluxo ordenado de eletrões fosse visto como água a correr numa mangueira, o conceito de **volt (V** - unidade da tensão elétrica) corresponderia à pressão/velocidade com que a água corre na mangueira e o **ampere (A** - unidade da intensidade da corrente elétrica) corresponderia à largura da mangueira, ou seja, a quantidade de água que pode existir numa determinada secção.

Se multiplicarmos a tensão pela intensidade da corrente, ou seja, Volt x Ampere, temos uma medida de trabalho conhecida como **watt - W**, que é a unidade geralmente utilizada para caracterizar a potência (P) dos dispositivos elétricos que temos em casa $1W = 1V \cdot 1A$.

Se tivermos uma torradeira que funciona com uma tensão de 220v (**volts**) e que tem uma potência (P) de 700 **watt**, sabemos que a intensidade (I) de corrente elétrica (**amperes**) que necessita para trabalhar é igual a $I = 700W / 220V = 3,18A^2$

Por fim, falta-nos mencionar o **Ohm (Ω)**, que é a unidade que define a resistência elétrica (R) que o sistema cria à passagem de corrente elétrica. Na analogia da mangueira de água, a resistência corresponderia a alguém a pisar a mangueira ou qualquer outra forma de atrito que a água encontrasse na sua passagem.

² Em situações ideais/teóricas, em que a eficiência do sistema é total (onde o fator potência = 1)

Todas estas unidades têm diferentes formas de ser apresentadas, dependendo das escalas em que nos encontremos a trabalhar. Por exemplo, a pequena luzinha que integra o interruptor do sôtão (falámos nosso no capítulo anterior) poderá consumir em torno de 0,02A, neste caso poderia utilizar-se o **miliampere - mA** para representar este valor: 20mA. Os dois são equivalentes.

No entanto é preciso cuidado, pois para utilizar as fórmulas matemáticas é necessário converter as unidades para o formato oficial do Sistema Internacional (ou seja: volt, ampere, watt e ohm).

A Tabela 1 apresenta alguns valores de conversão que poderão vir a ser úteis.

1	Ampere (A) é igual a	Conversão para A	Conversão de A
1000000	Microampere (µA)	Dividir por 1000000	Multiplicar por 1000000
1000	Miliampere (mA)	Dividir por 1000	Multiplicar por 1000
0,001	Kiloampere (kA)	Dividir por 0,001	Multiplicar por 0,001
1	Volt (V) é igual a	Conversão para V	Conversão de V
1000000	Microvolt (µV)	Dividir por 1000000	Multiplicar por 1000000
1000	Milivolt (mV)	Dividir por 1000	Multiplicar por 1000
0,001	Kilovolt (kV)	Dividir por 0,001	Multiplicar por 0,001
1	Watt (W) é igual a	Conversão para W	Conversão de W
1000	Mili-watt (mW)	Dividir por 1000	Multiplicar por 1000
0,001	Kilowatt (kW)	Dividir por 0,001	Multiplicar por 0,001
0,000001	Megawatt (MW)	Dividir por 0,000001	Multiplicar por 0,000001
1	Ohm (Ω) é igual a	Conversão para Ω	Conversão de Ω
1 000	Miliohm (mΩ)	Dividir por 1000	Multiplicar por 1000
0,001	Kiloohm (kΩ)	Dividir por 0,001	Multiplicar por 0,001

Tabela 1 - Tabela de conversão de unidades SI

Lei de Ohm

Existe uma fórmula matemática bastante simples que relaciona as 3 unidades acima descritas (V, A e Ω – o Watt é em si uma relação entre V e A), essa fórmula baseia-se na Lei de Ohm e diz-nos que a tensão elétrica (V) é igual à intensidade de corrente elétrica (I) multiplicada pela resistência elétrica (R), ou seja, **$V = I \cdot R$**

Uma mnemónica que ajuda a decorar esta fórmula é pensar nela como a fórmula VIRgem da eletricidade.

É através desta fórmula que conseguimos calcular o fluxo de eletricidade nos nossos circuitos e como ela se altera cada vez que encontra um componente no seu caminho. A Lei de Ohm é essencial para equilibrarmos o nosso sistema de modo a não o sobrecarregar (com perigo de fritar!) nem ficarmos aquém da energia necessária para o fazer funcionar na sua plenitude.

Ferramentas e componentes

Para colocarmos em prática alguns dos conceitos avançados até agora, vamos precisar de algum equipamento técnico.

Crocodilos

São pinças metálicas com isolamento que permitem agarrar e estabelecer contacto elétrico entre cabos e componentes. Geralmente estão ligados a cabos elétricos que contêm um crocodilo em cada uma das extremidades, permitindo ligar componentes de forma fácil e rápida, sem necessidade de utilizar técnicas complicadas ou outras ferramentas.



Figura 8 - Crocodilos com cabos (dois exemplares)

Multímetro

Como o próprio nome indica, um multímetro é um instrumento de medição (*metro*) que permite medir diversas grandezas (*multi*), neste caso elétricas. Por esse facto, é um instrumento que demora algum tempo a aprender. Neste manual

vamos apresentar as funcionalidades e modos de operação à medida que formos avançando na matéria.



Figura 9 - Multímetro

LED

O termo LED significa Light Emitting Diode (diodo emissor de luz). É um componente que emite luz quando atravessado por corrente elétrica. Uma vez que é um diodo, só admite passagem de corrente numa direção: do ânodo (recebe a corrente positiva na perna maior) para o cátodo (envia a corrente negativa pela perna menor). Cada LED é fabricado com um valor específico de tensão elétrica (V) e intensidade de corrente elétrica (A), ao ultrapassar estes valores de referência há uma forte probabilidade de “fritar” o LED ou reduzir-lhe drasticamente o seu tempo útil de vida.

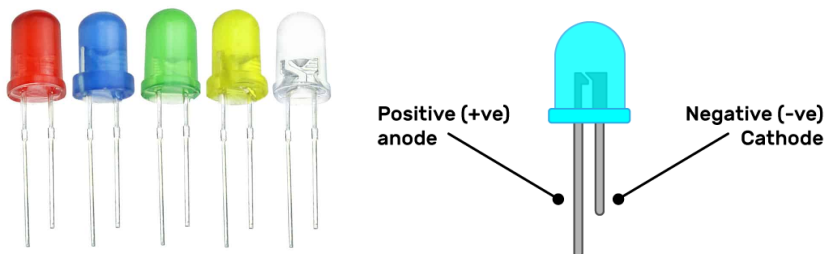


Figura 10 - Vários LEDs (esquerda); identificação do anodo e catodo (direita)

Exercício

Em grupos de ± 3 alunos vamos contruir uma pilha a partir de limões e peças metálicas e usá-la para acender um LED. Durante o processo vamos usar o multímetro para medir as diferentes grandezas elétricas no nosso circuito e compará-las com outras fontes de alimentação (elétrica) e outros circuitos.

1. Cortar um limão ao meio
2. Espetar um elemento de zinco e um elemento de cobre no limão
3. Medir a tensão elétrica (V) com o multímetro:
 - a. Selecionar a função V e ligar a ponteira preta no COM e a ponteira vermelha no V Ω Hz
 - b. Tocar com a ponteira vermelha no elemento de cobre e com a ponteira preta no elemento de zinco
4. Medir a intensidade da corrente elétrica:
 - a. Selecionar a função A e ligar a ponteira preta no COM e a ponteira vermelha no A
 - b. Tocar com a ponteira vermelha no elemento de cobre e com a ponteira preta no elemento de zinco
5. Com base nestas medidas e nas características do LED (V e A) adicionar – em série - a quantidade necessária de baterias de limão.
6. Usar o multímetro para fazer as mesmas medidas noutras fontes de alimentação (pilhas, transformadores).

C4. Introdução à eletrônica: circuitos elétricos em ambiente virtual

Sobre o capítulo

Objetivos de aprendizagem

- Conhecer as principais características e função de alguns componentes eletrônicos elementares (LED, interruptor, resistência. etc.);
- Identificar a diferenças entre ligação em série e ligação em paralelo;
- Consultar *data sheets* com vista à obtenção de informação detalhada sobre estes componentes;
- Aplicar de forma correta/otimizada estes componentes em circuitos elétricos de baixa complexidade;
- Implementar e analisar com ferramentas adequadas (multímetro) circuitos em ambiente de prototipagem virtual.

Conteúdo

- Apresentação de componentes eletrônicos elementares;
- Apresentação de *data sheets*;
- Introdução à prototipagem eletrônica em ambiente virtual;
- Exercícios práticos de prototipagem de eletrônica em ambiente virtual.

Materiais/equipamento

- Computador
- Software Tinkercad

Circuitos elétricos

Um circuito elétrico é composto por três elementos principais: uma fonte de energia que produza uma tensão elétrica (célula, bateria, tomada, painel solar, etc.), as cargas (*loads*) que transformam essa energia em outras formas de

energia (lâmpada, motor, resistência, etc.) e os elementos condutores que ligam os diferentes componentes (fios, cabos, tinta condutora, etc.)

Diagrama de circuito

O diagrama de circuito (*circuit diagram*) é a representação visual de um circuito eletrônico; é como um mapa do circuito que permite analisá-lo e partilhá-lo. Os diagramas podem ser pictóricos ou esquemáticos, dependendo se utilizam ilustrações dos componentes que compõe o circuito ou símbolos standardizados (Figura 11).

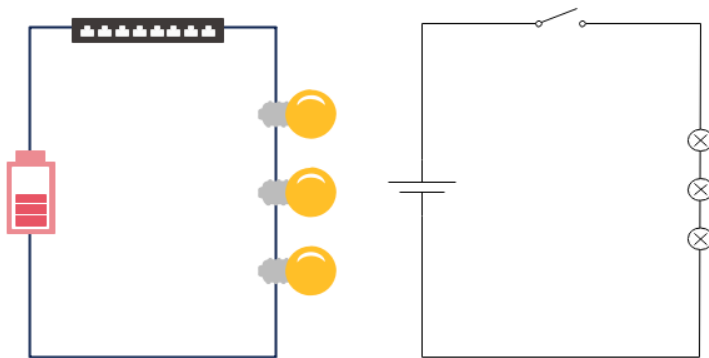


Figura 11 - Diagrama de circuito pictórico (esquerda) e diagrama de circuito esquemático (direita)

Para criação de diagramas de circuito são normalmente utilizados aplicações de software específicas para o design de circuitos eletrónicos (*EDA – Electronic Design Automation*). Alguns destes softwares não só permitem a criação de diagramas visuais como a sua simulação e análise em ambiente virtual.

Exemplos de aplicações para criação e simulação de diagramas de circuitos:

- Tinkercad Circuits - <https://www.tinkercad.com/learn/circuits>
- Kicad - <https://www.kicad.org/>
- Virtual Breadboard - <https://apps.microsoft.com/detail/9nblggh4tj3w>
- Circuit Lab - <https://www.circuitlab.com/editor/>
- Circuit Diagram - <https://www.circuit-diagram.org/editor/>
- Fritzing - <https://fritzing.org/download>

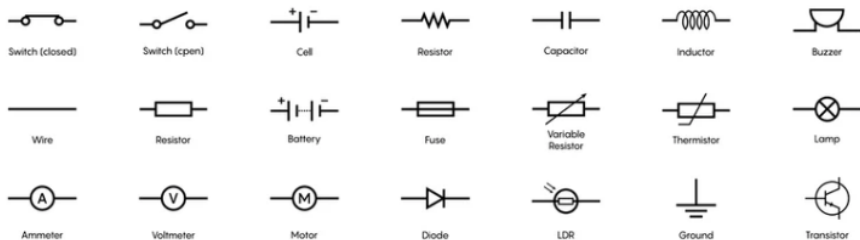


Figura 12 - Alguns símbolos padronizados usados na criação de diagramas.

Data Sheets

As *Data Sheets* (Figura 13) são os bilhetes de identidade dos componentes eletrônicos. São elas que nos dão informações essenciais sobre o componente a que se referem, nomeadamente a sua função e grandezas numéricas associadas (precisão do componente, parâmetros de corrente e tensão, etc.). As *data sheets* encontram-se disponíveis online e devem ser procuradas a partir do nome/marca do fabricante e referência do modelo em questão.

Rectangular LED

Hi Red, 2mm x 5mm

multicomp^{PRO}

RoHS Compliant

Specifications:

Die material	GaP
Emitted colour	Hi-Red
Lens colour	Red Diffused
Peak wavelength	630nm
Viewing angle	120°
Luminous intensity (IV)	1.2mcd

Absolute Maximum Ratings (T_a = 25°C)

Reverse Voltage	—
Reverse Current	10µA (V _R = 5V)
Operating Temperature Range	-40°C to 80°C
Storage Temperature Range	-40°C to 100°C
Lead Soldering Temperature Range	260°C for 5 Seconds
Exceed 1000 times per lot	

Electrical/Optical Characteristics at T_a = 25°C

Parameter	Symbol	Minimum	Typical	Maximum	Unit	Test
Luminous intensity	IV	0.5	1.2	1.8	mcd	I _F = 20mA
Viewing Angle	2θ 1/2	—	120	—	degrees	—
Peak Emission Wavelength	λP	—	630	—	nm	—
Dominant Wavelength	λD	—	630	—	nm	—
Spectral Line Half-Width	Δλ	—	20	—	nm	—
Forward Voltage	V _F	1.7	2.1	2.6	V	I _F = 20mA
Power Dissipation	P _D	—	—	45	mW	—
Peak Forward Current (Duty 1/10 at 10KHz)	I _F (Peak)	—	—	50	mA	—
Recommended Operating Current	I _F (Rms)	—	10	—	mA	—

Newark.com/multicomp-pro
Farnell.com/multicomp-pro
Element14.com/multicomp-pro

multicomp^{PRO}

Page <1>

17/08/19 V1.0

Figura 13 - Data Sheet para LED

Exercício

Parte 1

1. Replicar em ambiente virtual a bateria de limões feita na aula anterior.
2. Exportar um diagrama de circuito pictórico e outro esquemático.

Parte 2

1. Proceder à medição dos valores de corrente, tensão e resistência nos diferentes pontos do circuito.
2. Identificar os valores de corrente e tensão de funcionamento do LED.
3. Calcular a quantidade de limões necessária para acender um LED.
4. Adicionar os respectivos limões e simular.
5. Exportar o diagrama de circuito esquemático.

Circuito em paralelo e circuito em série

Uma das noções mais úteis e importantes na criação de circuitos é o conceito de ligação em **paralelo** e ligação em **série**. Tomando como exemplo o exercício da bateria de limões, esse circuito encontra-se em **série** pois os seus componentes encontram-se ligados sequencialmente, como se de um anel se tratasse. Existe apenas um único caminho para a eletricidade percorrer. Numa ligação em **paralelo**, o circuito apresenta ligações no interior do anel, tipo raios de uma roda, (Figura 14), o que significa que a eletricidade pode seguir diversos percursos.

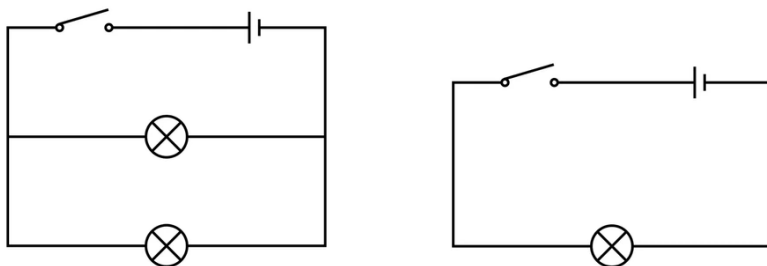


Figura 14 - Circuito em paralelo (esquerda) e circuito em série (direita)

Quando o circuito apresenta ligações em paralelo e ligações em série, é chamado um circuito misto. A tabela em baixo apresenta as principais diferenças entre circuito em paralelo e circuito em série.

Circuito paralelo	Circuito em série
A tensão mantém-se constante ao longo do circuito	A tensão pode variar ao longo do circuito se a resistência dos diversos componentes no circuito variar
A corrente pode variar ao longo do circuito se a resistência dos diversos componentes no circuito variar	A corrente mantém-se constante ao longo do circuito
Quando se somam resistências em paralelo a resistência total diminui	Quando se somam resistências em série a resistência total aumenta
Quando se interrompe o circuito num componente (por exemplo, um LED), o restante circuito continua a funcionar	Quando se interrompe o circuito num componente (por exemplo, um LED) o circuito deixa de funcionar porque o fluxo de corrente é interrompido

Queda de tensão

Uma das utilizações da Lei de Ohm que nos poderá ser bastante útil é no cálculo da queda de tensão num circuito elétrico em série, quando a corrente elétrica encontra resistência (uma luz, um motor, etc.). Como vimos na tabela em cima, a tensão (V - volts) pode variar ao longo do circuito, dependendo da resistência oferecida pelas diferentes cargas.

Nesse caso, para saber a diferença de potencial (ou seja, a tensão) em cada ponto específico do sistema, utiliza-se o valor da intensidade de corrente total do circuito (I) e substitui-se a resistência (R) do componente para o qual se pretende calcular a diferença de potencial.

Exercício

1. Alterar o circuito, colocando dois limões em paralelo.
2. Medir a corrente e tensão no circuito (usar dois multímetros).
3. Alterar os valores de tensão e resistência nos limões até o LED acender.

C5. Eletrônica analógica: circuitos eletrônicos com componentes físicos

Sobre o capítulo

Objetivos de aprendizagem

- Utilizar ferramentas de corte para cortar e descarnar cabo elétrico;
- Construir circuitos eletrônicos funcionais, em ambiente de prototipagem (*breadboard*), utilizando LEDs, resistências simples, resistências variáveis, interruptores, motores DC e várias fontes de alimentação;

Conteúdo

- Utilização de alicates
- Funcionamento da placa de ensaio (*breadboard*)
- Introdução aos motores DC, resistências variáveis e interruptores

Recursos

- Computador pessoal
- Alicates variados
- Placa de ensaio (*breadboard*)
- LEDs
- Resistências (simples, variáveis e fotossensíveis)
- Fio/cabo elétrico

Ferramentas e componentes

Para cumprir os objetivos de aprendizagem deste capítulo, vamos abordar alguma ferramentas e equipamentos novos.

Alicates

Os alicates são ferramentas essenciais quando se trabalha em eletrônica, pois permitem-nos cortar, apertar e descarnar fios elétricos. Existem diferentes tipos de alicates adequados aos diversos usos: alicate de pontas (com pinças finas para agarrar peças pequenas); alicate de corte (dedicado ao corte de fio); alicate universal (corta e aperta); alicate descarnador (com função de descarnar o fio, expondo o seu núcleo). Os alicates usados em eletrônica são geralmente mais pequenos que os alicates de uso geral.



Figura 15 - Alicate de corte (esquerda), alicate de pontas (centro) e alicate de universal (direita).

Placa de Ensaio (*breadboard*)

A placa de ensaio, mais conhecida por *breadboard*, é um equipamento destinado a testar ligações de forma fácil e rápida, sem a necessidade de utilizar solda, crocodilos ou outros meios de ligação.

A placa de ensaio é geralmente constituída por duas áreas distintas: as linhas da alimentação que correm longitudinalmente e as linhas de ligação que correm transversalmente.

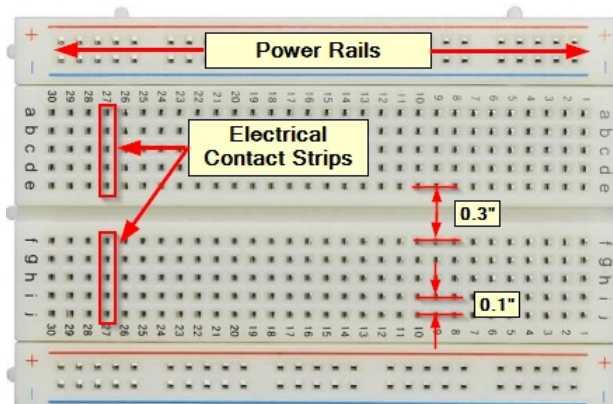


Figura 16 - Placa de ensaio (breadboard)

Resistências

As resistências são componentes eletrônicos passivos muito usados na criação de circuitos. A sua função é criar um entrave à passagem da eletricidade, sendo por isso medidas em Ohms (Ω).

As resistências são identificadas através de um código de cores que varia entre 3 e 6 tiras de cores, sendo as mais comuns as de 4 e 5 tiras (ver Figura 17). Outra forma de identificar a resistência em Ohms é medir diretamente com um multímetro ou recorrer a um calculador de resistências online³

Ao ligar resistências em paralelo, a resistência resultante será menor que a resistência com menor valor. Quanto mais forem adicionadas, mais diminui o valor de resistência total. Ao ligar resistências em série, estamos a aumentar o valor de resistência, correspondendo este à soma aritmética de todas as resistências. O valor da resistência resultante, quer em paralelo quer em série, também pode ser calculado online⁴.

Um dos usos mais frequentes para as resistências está relacionado com o controlo do fluxo de eletricidade na alimentação de equipamento sensível como os LEDs. Através da queda de tensão, é possível usar fontes de alimentação relativamente potentes com componentes de menor tensão e corrente.

³ <https://www.calculator.net/resistor-calculator.html>

⁴ <https://www.digikey.pt/en/resources/conversion-calculators/conversion-calculator-parallel-and-series-resistor>

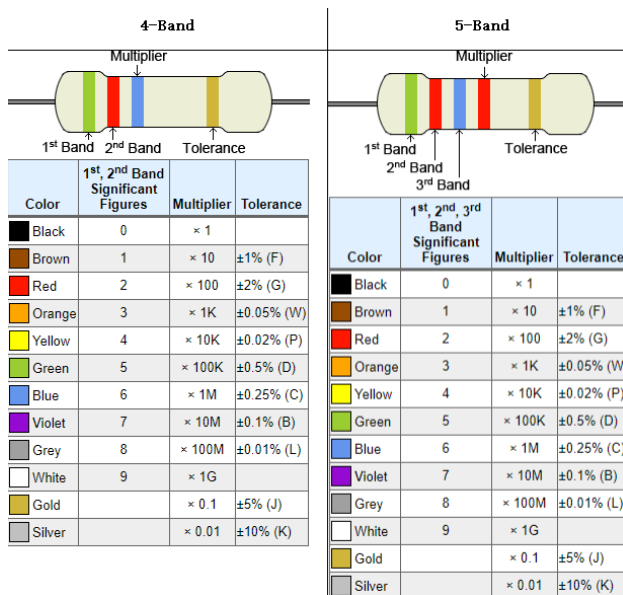


Figura 17 - Código de cores para identificação das resistências

Existem resistências cujo valor não é fixo, são chamadas de resistências variáveis. A escolha do valor de resistência pode acontecer, por exemplo, através da quantidade de luz a incidir na resistência (foto resistência) ou através de um controle giratório (potenciômetro).

Atenção que as resistências estão preparadas trabalhar com determinados limites de potência (em watts – consultar o valor na *data sheet*). Ao exceder essa potência, a resistência “frita”!

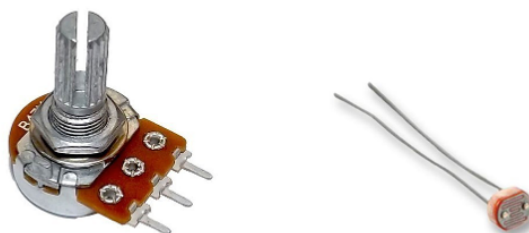


Figura 18 - Potenciômetro (esquerda) e foto resistência (direita)

Motor DC

Como o próprio nome indica, um motor DC funciona com corrente contínua. Através da criação de um campo eletromagnético, o motor permite converter energia elétrica em energia mecânica de rotação e vice-versa. Os motores DC funcionam geralmente numa gama de tensão, podendo alterar a sua velocidade de rotação (rpm – rotações por minuto) através da alteração da tensão.



Figura 19 - Motor DC - 1,5V a 3V - 16000rpm.

Interruptor

Como o próprio nome indica, um interruptor serve para interromper um circuito, permitindo assim que este funcione apenas quando se pretende. Tem geralmente a forma de um botão ou pequeno manípulo. Existem diversos tipos de interruptores: simples, comutadores, momentâneos, permanentes, etc.

Exercício

O objetivo do exercício é acender um LED sem o danificar, usando as fontes de alimentação fornecidas e resistências.

1. Identificar a voltagem e corrente de funcionamento do LED (consultar a *data sheet* ou usar a fonte de alimentação de laboratório);
2. Calcular o valor da resistência;
3. Fazer as ligações na placa de ensaio, colocando um interruptor.

Repetir o exercício usando o motor DC. Utilizar um potenciômetro e/ou foto resistência para alterar a sua velocidade de rotação.

C6. Eletrónica digital: introdução aos microcontroladores

Sobre o capítulo

Objetivos de aprendizagem

- Conhecer as principais características e funcionalidades de um microcontrolador;
- Identificar as marcas mais relevantes de microcontroladores existentes no mercado DIY;
- Instalar e utilizar o IDE Arduino com diferentes microcontroladores;
- Escrever e carregar código elementar para programar um microcontrolador.

Conteúdos

- Introdução aos microcontroladores;
- Introdução à linguagem de programação Arduino e respetivo IDE;
- Exercício prático: Acender o LED.

Recursos necessários

- Computador (com software Arduino IDE)
- Microcontroladores (Arduino) + cabo USB
- LED
- *Breadboard*
- Cabos para *breadboard*

Microcontroladores

Definição

Os microcontroladores (MCU – *microcontroller unit*) são dispositivos eletrônicos de controlo que tipicamente incorporam num só chip uma unidade de processamento (CPU), memória, I/O e conversão A/D e D/A.

Ao contrário de um computador tradicional que permite a instalação de múltiplas aplicações (através do seu sistema operativo), o microcontrolador é normalmente utilizado para executar um número limitado de funções específicas. Além disso, o seu consumo energético e preço são extremamente baixos quando comparados com um computador tradicional (inclusive com os microcomputadores como o Raspberry Pi).

Embora o conceito de microcontrolador se reporte especificamente às características presentes num chip, é comum que esse chip seja vendido numa placa de circuito (normalmente chamada de módulo) que facilita a ligação e interoperabilidade entre os vários componentes do chip e os periféricos (fontes de alimentação, sensores, computador, etc.).

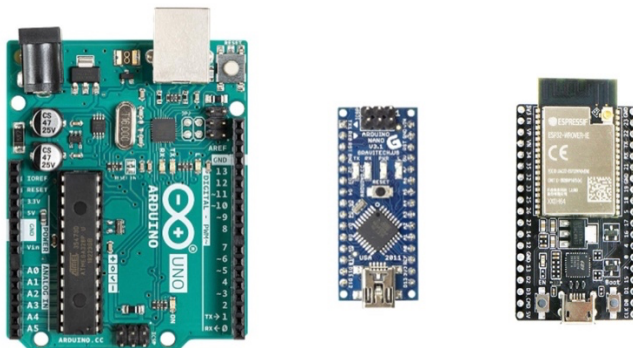


Figura 20 - Diferentes módulos de microcontroladores (da esquerda para a direita): Arduino UNO, Arduino nano, ESP32.

Tipos de microcontroladores

Existem múltiplas tipologias de microcontroladores - geralmente dependentes dos seus propósitos e fabricantes – cujas características variam ao nível do

número de bits (8bits, 16bits, 32bits, etc.) e arquitetura (ARM, PIC, AVR, FPGA, Xtensa, etc.).

Os microcontroladores geralmente utilizados no universo DIY, em particular na área da arte interativa, são os microcontroladores ARM 8bits, como por exemplo o ATmega328P, presente no módulo Arduino Uno, e os microcontroladores Xtensa 32bits, presentes na família de módulos ESP32 desenvolvidos pela Espressive Systems.

Anatomia de um microcontrolador

Embora as características de um microcontrolador estejam amplamente dependentes do seu modelo/fabricante, existem elementos que se podem encontrar na generalidade dos microcontroladores. A título de exemplo, vamos usar o módulo Arduino Uno para demonstrar a localização destes elementos.

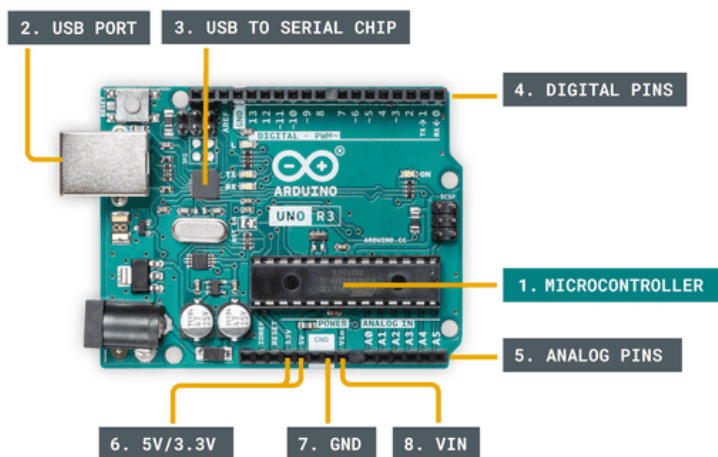
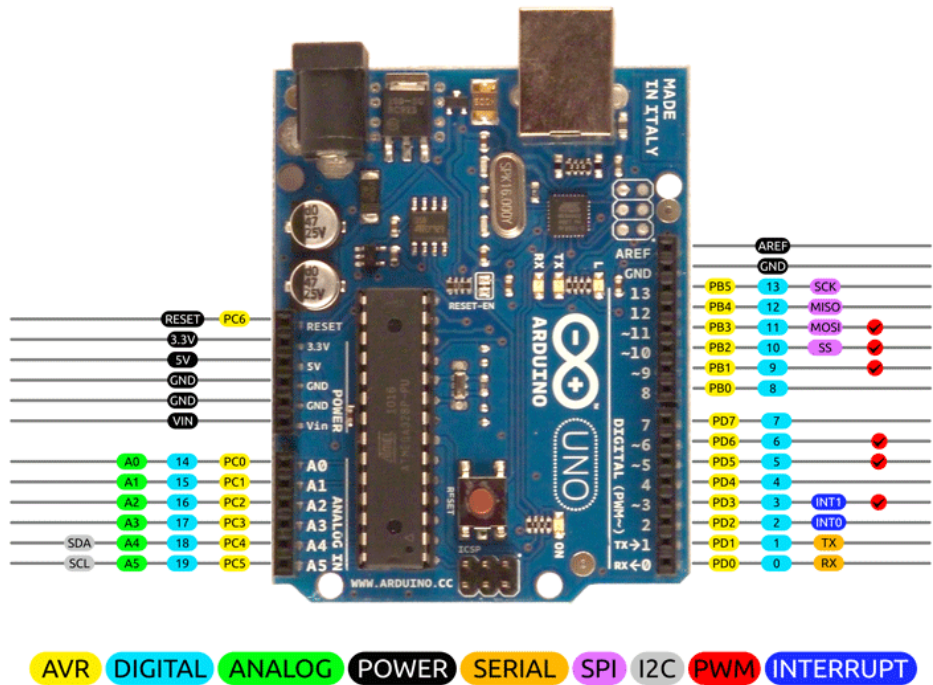


Figura 21 - Anatomia de um módulo Arduino Uno (retirado de <https://docs.arduino.cc/>)

1. Microcontrolador – o chip que é cérebro do módulo;
2. Porta USB - utilizada para ligar o módulo Arduino a um computador;
3. Chip USB para Série – chip dedicado a traduzir dados de USB para série;
4. Pinos digitais - utilizam lógica digital (0/1). Normalmente utilizados para interruptores. Funcionam com 20mA de corrente (entrada e saída) e 5V;

5. Pinos analógicos - pinos que podem ler valores analógicos a uma resolução de 10 bits (0-1023);
6. Pinos de 5 V / 3,3 V - pinos usados para alimentar componentes externos;
7. GND – terra/negativo, usado para completar um circuito;
8. VIN - Voltage In, pino onde se ligam fontes de alimentação externas.

Quanto à função específica de cada pino (alguns conhecidos por GPIOs: General-Purpose Input/Output), é útil socorrermos-nos do seguinte diagrama:



2014 by Bouni
Photo by Arduino.cc

Figura 22 - Diagrama de pinos (pinout diagram) do Arduino UNO

Protocolos de comunicação em série/serial

Os microcontroladores tem a capacidade (necessidade!) de comunicar entre si e com outros dispositivos, como sensores, computadores, etc. Os protocolos de comunicação em série (ou serial) são fundamentais nesta tarefa, baseando-se na transmissão de dados digitais (bits) – ou seja, zeros e uns – sequencialmente e de forma muito rápida e robusta. Os protocolos de comunicação em série mais comuns no mundo dos microcontroladores são o UART (Universal Asynchronous Receiver-Transmitter), SPI (Serial Peripheral Interface) e I2C (Inter-Integrated Circuit). Na maioria dos módulos Arduino, o UART é responsável pela na comunicação entre o módulo e o computador, quer seja para carregar novos programas ou para ler dados diretamente do módulo. No caso do Arduino UNO, o protocolo UART é “traduzido” para outro protocolo de comunicação em série muito conhecido, o USB (Universal Serial Bus).

Embora o protocolo UART seja muito comum e simples de utilizar, ele é considerado antiquado e, como tal, limitado quando comparado com os outros. Por exemplo, o UART só permite ligações um-para-um, enquanto que os protocolos I2C e SPI permitem ligar vários dispositivos à mesma porta.

Outra diferença importante é a taxa de transferência de informação: enquanto que o UART funciona em velocidades na ordem do 100.000Hz (chamada *baud rate*), o SPI consegue velocidades na ordem dos 10MHz (10.000.000Hz).

Alimentar um microcontrolador

Para trabalhar com um microcontrolador é essencial que este seja alimentado com corrente elétrica adequada. Cada módulo tem as suas necessidades energéticas e modo(s) de alimentação específicos, e por isso é essencial recorrer aos respetivos manuais para não errar!

No caso do Arduino Uno, a alimentação pode ocorrer de 4 formas diferentes (nunca usar mais do que uma forma de alimentação):

- USB Type B 2.0 5V (fonte regulada)
- Transformador 7-12V, 1A
- VIN pins 7-12V (nunca inverter polaridade!)
- 3V3/5V output pins 3 ou 5V (não exceder!) – *não aconselhável!*

No caso do ESP-WROOM-32, a alimentação pode ocorrer de 3 formas específicas (nunca usar mais do que uma forma de alimentação!):

- USB Micro B 5V (fonte regulada)
- 5V pin 5-12V ($\pm 6V$); 0,5A
- 3V output pin 3V (fonte regulada) – não aconselhável!

Programação de microcontroladores

Para definir um conjunto de funções/tarefas de um microcontrolador é necessário recorrer a uma linguagem de programação informática. É comum que um microcontrolador admita ser programado com mais do que uma linguagem de programação, como é o caso dos módulos ESP32 que podem ser programados com Espressif IDF, Arduino C/C++, Micropython, JavaScript, LUA, entre outros.

Arduino IDE – Instalação

No contexto deste manual vamos fazer uma introdução à programação de microcontroladores recorrendo à linguagem de programação Arduino (baseada em C/C++) e ao seu IDE (*Integrated Development Environment*), que na prática nos oferece uma aplicação muito conveniente (e gratuita!) para escrita de código, *debugging*, *upload*, monitorização, comunicação, etc.

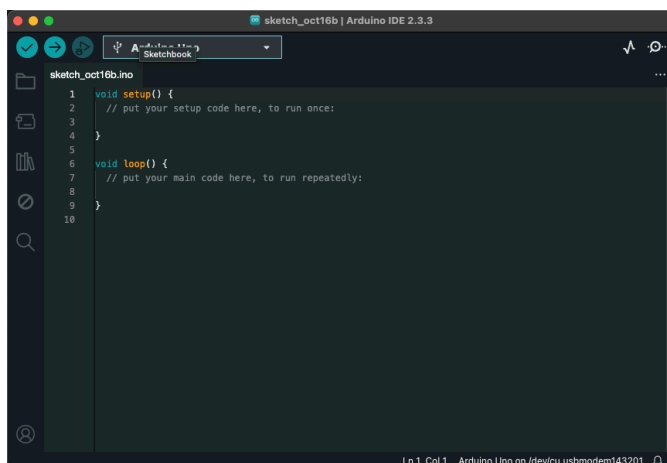


Figura 23 - Arduino IDE

O Arduino IDE existe para Windows, Mac e Linux, e deve ser descarregado aqui: <https://www.arduino.cc/en/software>

Depois de descarregado (descarregar a última versão, mas evitar os *Nightly Builds* – versões beta não totalmente testadas/estáveis), proceder à instalação no computador que será usado para escrever o código a carregar para o microcontrolador.

Existe também uma versão online do Arduino IDE, chamado Arduino Cloud, que pode ser acedido aqui: <https://app.arduino.cc/>. No contexto deste manual, será adotado o Arduino IDE.

Depois de instalado no computador, deve ligar-se o módulo ao computador (via USB) e abrir-se o Arduino IDE – para exemplificação, iremos usar o Arduino UNO.

Selecionar o módulo e a porta de série (*serial*) no IDE (*dropdown menu*). Caso não esteja disponível, procurá-lo em “*select other board and port...*”.

Arduino API (aka Linguagem de Programação Arduino)

Esta secção é dedicada à Arduino API (API significa *Application Programming Interface*), normalmente chamada de linguagem de programação Arduino. Aqui procura-se apenas identificar os elementos básicos específicos desta linguagem, permitindo ao programador inexperiente adquirir uma noção geral da estrutura e abordagens à programação em linguagem de programação Arduino. Não se pretende dar os fundamentos da programação, pois tal empreendimento sai fora dos objetivos desta unidade curricular e, consequentemente, deste manual.

Como foi referido, a linguagem de programação Arduino é baseada na popular linguagem de programação C++, uma linguagem orientada a objetos bastante robusta e amplamente utilizada no universo das ciências informáticas. A sua curva de aprendizagem é um pouco íngreme, mas os resultados valem o esforço.

A linguagem de programação Arduino encontra-se dividida em três partes: funções (controlam o Arduino executam operações computacionais), valores (variáveis/constantes e tipos de dados) e estrutura.

Sketch

Cada programa criado na Arduino API é chamado de *sketch* (esboço). Para funcionar, um *sketch* tem de ter, no mínimo, duas funções obrigatórias que

constituem a estrutura básica da programação em Arduino: `void setup()` e `void loop()`;

- `void setup()`: é onde se faz o *setup* do programa, ou seja, as configurações iniciais que sabemos ser fundamentais para o arranque do programa.
- `void loop()`: após a configuração inicial (*setup*), o programa entra na função *loop* (repetição) onde se encontram, de facto, as suas instruções mais importantes. Como o próprio nome indica, esta função está preparada para funcionar em *loop* (a uma frequência de repetição de várias dezenas de vezes por segundo, dependendo do microcontrolador e complexidade do código), o que significa que o código se repete infinitamente até que o programa *crash*, falte a energia ou receba uma ordem para terminar.

No fim de criado o programa, este é carregado (*uploaded*) para o microcontrolador, começando imediatamente em execução assim que o Arduino estiver ligado à corrente. Os *sketches* são ficheiros com extensão `.ino`.

Elementos específicos da Arduino API

Embora muitos dos elementos da linguagem de programação Arduino sejam comuns a outras linguagens de programação (e por isso facilmente reconhecíveis por programadores experientes), existem alguns comandos que são específicos do Arduino. Apresentamos de seguida alguns dos mais relevantes:

Pinos digitais

- `digitalRead()` – retorna o estado de um pino digital (1/0)
- `digitalWrite()` – define o estado de um pino digital (1/0)
- `pinMode()` – define o modo de funcionamento de um pin digital

Pinos analógicos

- `analogRead()` – retorna o valor de um pin analógico (10bit: 0-1023)
- `analogReadResolution()` – define a resolução da conversão A/D (int)
- `analogReference()` – define a voltagem de referência do módulo (int)
- `analogWrite()` – define valor num pino analógico c/ PWM (8bit: 0-255)
- `analogWriteResolution()` – define a resolução de escrita (int)

Constantes

- `HIGH`; `LOW` – estado de um pino: 1 e 0
- `INPUT`; `OUTPUT`; `INPUT_PULLUP`; `INPUT_PULLDOWN` – Modo dos pinos
- `LED_BUILTIN` – representa o led do módulo

Para mais informações sobre estes e outros comandos da linguagem de programação Arduino, consultar a referência: <https://docs.arduino.cc/language-reference/>

Bibliotecas

A linguagem de programação de base do Arduino pode ser expandida através da adição de novos comandos para utilizações específicas, como por exemplo um determinado hardware. Para procurar e instalar uma biblioteca a partir do Arduino IDE seguir os seguintes passos:

- Abrir o painel de Gestão de Bibliotecas (Tools > Manage Libraries);
- Pesquisar o nome da biblioteca pretendida;
- Clicar o botão de “*Install*” na biblioteca pretendida;
- No início do sketch/código, colocar a expressão `#include <biblioteca>`.

Exemplos práticos

Tradicionalmente, a introdução à programação – independentemente da linguagem em questão - tem como ponto de partida a criação do programa “Hello World”. Este programa seminal, tem como objetivo introduzir o aprendiz à estrutura mais básica e elementar que uma determinada linguagem de programação pode ter.

No caso da Arduino API, o programa Hello World consiste na escrita dessas duas palavras na janela de monitorização serial (*Serial Monitor*). Considerando a importância deste protocolo de comunicação no universo Arduino, este pequeno programa acaba por ser bastante útil.

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

Analisando as funções que aparecem dentro das estruturas `setup()` e `loop()`, temos:

- `Serial.begin(9600)`: Inicia uma ligação em série, indicando a taxa de transferência de bits na unidade bauds. Neste caso: 9600 bauds.
- `Serial.println("Hello world!")`: Imprime na janela de monitorização serial as palavras Hello World. De referir que é necessária a utilização de aspas.

C7. Eletrônica digital: atuadores

Sobre o capítulo

Objetivos de aprendizagem

- Ligar e controlar atuadores com um microcontrolador, nomeadamente LEDs, Buzzers e motores DC;
- Utilizar funções específicas para determinados atuadores (buzzers);
- Controlar atuadores com sinais digitais HIGH/LOW e PWM;
- Controlar atuadores com sinais analógicos;
- Controlar *switches/drivers* que por sua vez controlam outros atuadores.

Conteúdo

- Introdução aos atuadores;
- Exercícios práticos com atuadores e drivers.

Recursos necessários

- Computador (com software Arduino IDE)
- Microcontroladores (Arduino) + cabo USB
- 2 LEDs
- Transístor MOSFET
- Resistências (100Ω e 220Ω)
- Motor DC
- Fonte de alimentação externa
- Besouro (buzzer)
- *Breadboard*
- Cabos para *breadboard*

Sobre atuadores e microcontroladores

Neste capítulo vamos aprender a ligar e controlar atuadores com um microcontrolador. Em primeiro lugar importa esclarecer, ainda que de forma relativamente simplista, que um atuador é um dispositivo que altera um estado físico através da energia que recebe, estamos a falar de LEDs (transformam energia elétrica em luz e calor), motores (transformam energia elétrica em movimento), besouros (transformam energia elétrica em som), entre outros.

Alguns (poucos!) atuadores podem ser ligados diretamente ao microcontrolador, utilizando a energia dos pinos. Como referido anteriormente, no Arduino, os pinos digitais debitam 20mA de corrente (entrada e saída) e 5V de tensão. Isso servirá para acender LEDs de 5mm e pouco mais.

O pino de saída 5V pode debitar até cerca de 500mA quando o Arduino é alimentado pela porta USB do computador, ou um pouco mais se alimentado por um transformador. Em todo o caso, não devemos pensar o Arduino como uma fonte energética, mas apenas como um dispositivos de controlo. Existe o risco (muito) real de fritarmos o microcontrolador se insistirmos em alimentar dispositivos com ele.

Relativamente ao tipo de controlo que podemos ter, vamos explorar principalmente 3 modos: o High/Low dos pinos digitais, o PWM (pulse with modulation) dos pinos digitais – úteis para altera a intensidade de LEDs – e pins analógicos.

Exercícios práticos

LED do módulo a piscar

Este exemplo, retirado dos exemplos do Arduino IDE (*File > Examples > Basic > Blink*), controla o piscar do LED que está presente no módulo. Neste caso não ligamos nada ao módulo (apenas o cabo USB).

```

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(2000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(500);
}

```

💡 Experimenta a alterar os valores do `delay()` para veres o resultado!

LED externo a piscar

Para colocar um LED externo a piscar, é apenas necessário criar um pequeno circuito igual ao da imagem. O fio preto liga ao GND (terra) e o vermelho ao PIN 13 (digital). Para não danificar o LED ou o Arduino, teremos de usar uma resistência entre o PIN 13 e ânodo. Antes de fazer qualquer ligação, desligar o cabo USB!

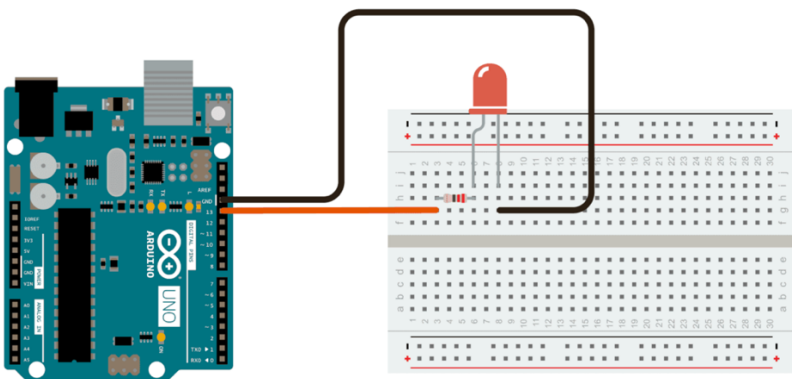


Figura 24 - Circuito para LED exterior (retirado do site arduino.cc)

O cálculo para achar uma resistência funcional é feito com recurso à lei de Ohm. Sabemos que o LED funciona com uma tensão de $\pm 2V$ e uma corrente de 20mA. Se o PIN 13 tem uma tensão de 5V, a diferença para a tensão do LED é de 3V. Substituindo na fórmula:

$$V = I \cdot R \Leftrightarrow R = \frac{3V}{0,02A} \Leftrightarrow R = 150\Omega$$

Por precaução e facilidade, usemos uma resistência um pouco maior, de 220 Ω .

Neste caso, não precisamos de refazer o código, pois o PIN 13 é representado pela constante `LED_BUILTIN`.

Dois LEDs externos a piscar

Neste exercício vamos adicionar um led externo a piscar intercaladamente com o existente. Para isso repetimos o circuito, adicionando um LED e uma resistência. O novo LED deve partilhar a ligação à terra do LED anterior e a resistência deve ser ligada ao PIN 12.

Em termos do código vamos introduzir uma novidade: as variáveis!

As variáveis são elementos fundamentais na programação informática. Elas permitem-nos associar um nome escolhido por nós a um determinado conteúdo “variável” (pode ser um número, uma letra, uma imagem, etc., dependendo tipo de variável que escolhemos).

Neste caso, vamos associar a palavra `segundoLed` ao número 12, algo que fazemos antes da função `void setup()`, pois assim esta variável torna-se global, podendo ser usada em qualquer parte do programa.

Uma vez que a variável é um número inteiro (neste caso, o 12), temos de indicar que se trata de uma variável do tipo `int` (se fosse 12,5 seria um número decimal, logo a variável seria do tipo `float`).

```
int segundoLED = 12;

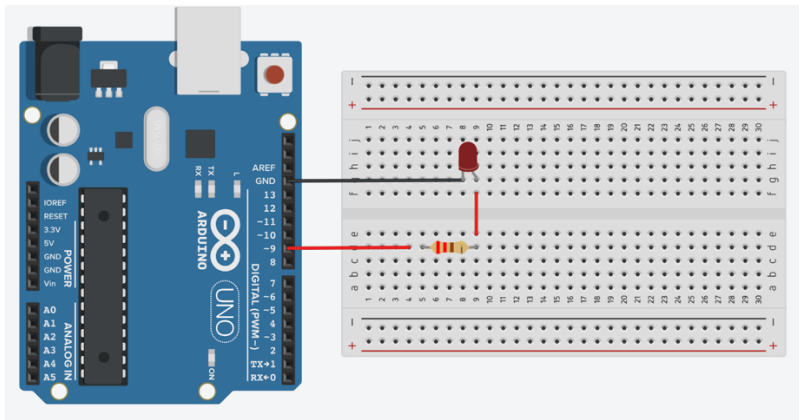
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(segundoLED, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  digitalWrite(segundoLED, LOW);
  delay(2000);
  digitalWrite(LED_BUILTIN, LOW);
  digitalWrite(segundoLED, HIGH);
  delay(500);
}
```

Alterar a intensidade de um LED

Neste exercício vamos alterar a intensidade de um LED usando impulsos digitais com a técnica de PWM (Pulse Width Modulation ou Modelação por Largura de Pulso). Como vimos anteriormente, um pino digital só tem dois estados: High e Low (acesso e apagado). Para conseguirmos fazer com que um LED diminua a sua intensidade, temos de usar a técnica de PWM que consiste em apagar e acender o LED de forma tão rápida que nos dá a sensação de que o LED está a alterar a sua intensidade. O PWM é utilizado através da função `analogWrite()` e só pode ser utilizado em pinos específicos, que permitam essa função. No Arduino esses pinos estão assinalados com o símbolo ~.

Em termos do circuito, teremos fazer uma pequena alteração ao circuito do exercício *LED externo a piscar*, ligando o ânodo ao pino 9.



Quanto ao código, vamos utilizar o seguinte:

```
int pinLed = 9;

void setup() {
  pinMode(pinLed, OUTPUT);
}

void loop() {
  analogWrite(pinLed, 255);
  delay(1000);
  analogWrite(pinLed, 10);
  delay(1000);
}
```

💡 Experimenta a criar vários padrões de intensidades.

O besouro a afinar

Agora vamos experimentar com um dispositivo output sonoro. Neste caso vamos ligar um besouro, ou *buzzer* em inglês. Trata-se de um piezo, um dispositivo que tanto pode ser usado como microfone ou coluna.

Para ligar o besouro, vamos fazer as seguintes ligações: perna negativa do besouro à terra no Arduino, perna positiva a uma resistência de 100Ω , seguida de ligação ao PIN 9.

Estas ligações devem ser executadas na *breadboard* (sempre desligada da alimentação USB!).

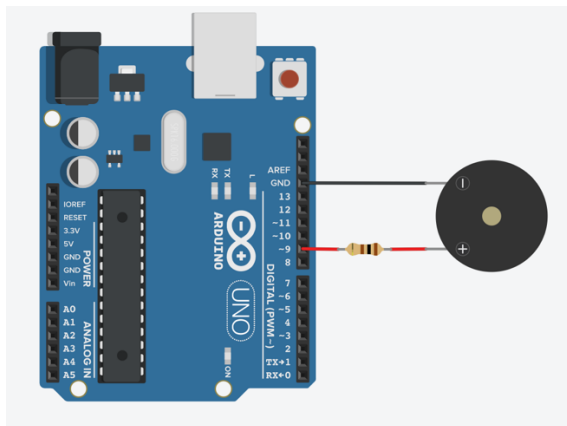


Figura 25 - Ligação de besouro.

Para fazer o besouro tocar temos de usar a função `tone()`. Esta função admite três parâmetros: número do pin, frequência do som em Hz e a duração do som em milissegundos (opcional). Para terminar um som que não tenha duração definida, usar a função `noTone()`. Acrescentar este código ao anterior.

```
int buzzerPin = 9;

void setup() {
}

void loop() {
    tone(buzzerPin, 1000);
    delay (100);
    noTone(buzzerPin);
    delay (100);
}
```

💡 Experimenta a alterar os valores da frequência para ouvires o resultado!

O besouro a cantar

Neste exercício vamos fazer o besouro cantar um *sweep*, ou seja, um varrimento de frequências do mais grave para o mais agudo.

Para fazer isso vamos introduzir uma novidade no código: a estrutura de controlo chamada ciclo `for()`. Este ciclo permite repetir uma determinada parte do código por um número pré-determinado de vezes. Em português corrente, é o mesmo que dizer: faz esta tarefa x vezes.

A forma de escrever um ciclo `for()` é a seguinte:

```
for (int i = 1; i < x; i = i + 1){
    // colocar aqui o que se pretende repetir
};
```

`int i = 1` → inicializa a variável `i` com o valor 1, ou seja, o `i` passa a ter o valor 1;
`i < x` → isso acontece enquanto `i` for menor que `x` (valor que temos de indicar);
`i = i + 1` → em cada ciclo, a variável `i` é incrementada um valor.

Voltando ao código do besouro a cantar, vamos fazer com que o nosso besouro cante da frequência 100 à frequência 1000. Em vez de escrevermos 900 linhas de código, podemos simplesmente usar o ciclo `for()` dentro da função `void loop()`.

```
for (int i = 100; i < 1000; i = i + 1){  
  tone (buzzerPin, i);  
  delay (5);  
};
```



Experimenta alterar a frequência e `delay()` para ouvires o resultado!

Exercício

Partindo dos conhecimentos adquiridos até aqui, constrói um circuito e escreve o código que permita um LED aumentar e diminuir a sua intensidade com um padrão orgânico, que simule a respiração de um ser humano.

Controlar atuadores que contenham a sua fonte de alimentação

Como referido anteriormente, um microcontrolador não deve ser entendido como uma fonte de alimentação, ou seja, não deveremos alimentar componentes com a corrente que sai do microcontrolador, a menos que o componente consuma muito pouca energia (como os LEDs de 5mm que temos vindo a utilizar).

No caso geral de atuadores que necessitem de corrente elétrica considerável (aproximadamente superior a 20mA), teremos de utilizar um componente

eletrónico intermédio que recebe sinal do microcontrolador para, em seu turno, controlar o referido atuador alimentado por uma fonte de alimentação externa.

Segue-se uma descrição de 3 componentes que são geralmente utilizados nestas situações: o transístor, o relé e o *driver*.

Transístor

O transístor é um componente semicondutor que transita entre estados, ou seja, transita entre o estado de condutor (deixa passar corrente elétrica) e não-condutor (bloqueia a passagem de corrente elétrica), sendo essa transição efetuada a partir das ordens que recebe, por exemplo, do microcontrolador. Um facto curioso é que a referida transição não é obrigatoriamente binária (aberto/fechado), ou seja, poderão existir estados intermédios em que passa mais ou menos corrente.

Um transístor é geralmente constituído por três terminais, sendo que num dos terminais entra o sinal de controlo (que sai do microcontrolador) e nos outros passa a corrente que irá para o atuador. Usando a analogia da água, o transístor é uma torneira em que nós (o microcontrolador) rodamos/controlamos o manípulo, controlando assim a corrente de água que vai do tubo que está na parede (fonte de alimentação externa) para o copo que queremos encher (atuador).

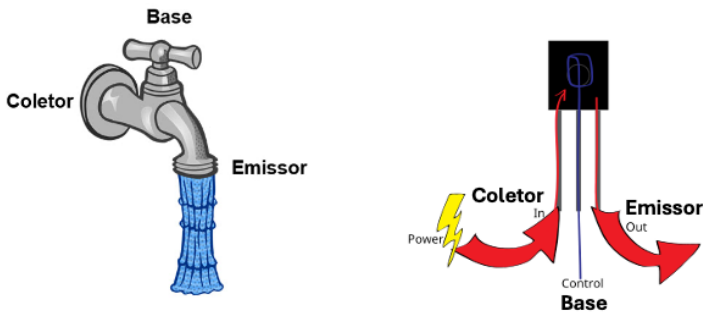


Figura 26 – À esquerda: analogia do funcionamento de um transístor (retirado do site ptrobotics.com). À direita: esquema de funcionamento de um transístor (adaptado do site wikipedia.org). Aplicável a transístores BJT do tipo NPN.

Os terminais de um transístor têm os seguintes nomes:

- **Source (FET) / Emissor (BJT):** É aqui que entra a corrente proveniente de uma fonte de energia externa (nos MOSFTE *n-channel* é invertido).
- **Gate (FET) / Base (BJT):** Aqui entra a corrente proveniente do microcontrolador, a qual irá alterar o estado de condutividade do transistor. É uma corrente de baixa tensão ($\pm 5V$);
- **Drain (FET) / Coletor (BJT):** Desta perna sai a corrente que irá para o atuador, a qual é proveniente da fonte de energia ligada ao emissor (nos MOSFTE *n-channel* é invertido).

Existem vários tipos de transístores, sendo que dois dos tipos mais comuns são os BJT (*bipolar junction transistor*) e os FET (*field-effect transistor*).

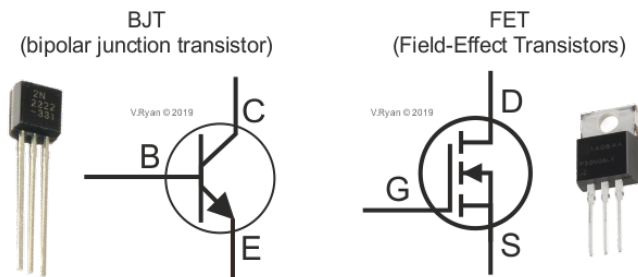


Figura 27 - Diferenças entre BJT e FET (retirado de technologystudent.com)

Embora a sua função e funcionamento seja semelhante, existem algumas diferenças entre eles:

- **Nomenclatura:** os terminais têm nomes diferentes (ver em baixo);
- **Velocidade:** a transição entre estados é mais rápida nos FET;
- **Consumo:** os FET requerem menos potência para operar;
- **Temperatura:** os FET são mais estáveis em relação à temperatura;
- **Ruído:** os FET são mais imunes ao ruído elétrico;
- **Durabilidade:** Os BJT são mais duráveis;
- **Preço:** os BJT são mais baratos;
- **Controlo:** os BJT são controlados com corrente, os FET com tensão;

- **Complexidade de utilização:** Os BJT são mais permissivos em relação à corrente utilizada para controlo;
- **Utilização:** Os BJT são geralmente utilizados em circuitos de baixa potência, enquanto os FET são utilizados em circuitos de maior potência.

Além destas diferenças, cada um dos tipos de transístor apresentados divide-se em várias subcategorias, sendo que as mais relevantes no contexto deste manual referem-se às que são dependentes da polaridade da corrente utilizada para controlo. Assim, quando usamos tensão positiva na base/*gate*, estamos perante transístores NPN (BJT) e N-*channel* (FET)⁵, quando usamos tensão negativa na base estamos perante PNP (BJT) e P-*channel* (FET).

Estas e outras características devem ser sempre consultadas na ficha técnica (*datasheet*), nomeadamente os valores de tensão/corrente permitidos em cada terminal, os quais são apresentados nos seguintes rácios, como por exemplo:

$$\beta(\text{current gain}) = \frac{\text{collector current}}{\text{base current}}$$

Relé

O relé tem uma função semelhante à do transístor, mas a sua forma de construção não lhe permite estados intermédios (é baseado em tecnologia eletromecânica e não semicondutora). O relé ou está fechado ou aberto; ou permite passagem de energia ou a bloqueia. Além disso, a grande diferença entre o funcionamento do transístor e do relé está relacionada com a quantidade de corrente que permite controlar, sendo este utilizado na comutação de maiores potências (por exemplo, 220v AC).

Os relés são baseados numa bobine que quando é energizada (com uma voltagem baixa $\pm 5V$) cria um campo eletromagnético que liga/desliga um interruptor no seu interior.

⁵ Existem vários tipos de FET transístores, sendo os MOSFET um dos mais utilizados. Neste caso, estamos a considerar um MOSFET do tipo *enhancement mode*, o que significa que quando não recebe corrente no terminal *gate*, não passa corrente entre os terminais *source* e *drain*. Os MOSFET em *depletion mode* passam corrente sempre que não recebam qualquer tensão no terminal *gate*, e bloqueiam a sua passagem quando essa tensão existe.

Um relé tem normalmente entre 4 e 5 terminais: 2 dedicados a energizar a bobine, e 2/3 dedicados à passagem/corte da corrente que se pretende controlar. Os relés de 5 terminais permitem configurar sistemas sempre abertos (*no* – *normally open*), bloqueando a passagem de corrente quando não recebem corrente de controlo, e sempre fechados (*nc* – *normally closed*), permitindo a passagem de corrente quando não recebem corrente de controlo.



Figura 28 - Relé com 5 terminais.

Para conhecer as especificações de cada relé, será sempre necessário consultar a respetiva ficha técnica (*datasheet*).

Drivers

O termo *driver* é aqui apresentado como uma tipologia genérica de circuitos com funções de controlo específicas para cada tipo de atuador. Por exemplo, existem *drivers* específicos para LEDs, Motores DC, Piezos, etc. Alguns dos *drivers* têm bibliotecas associadas que facilitam muito a sua programação.

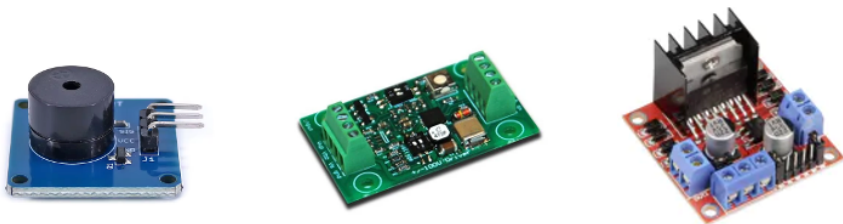


Figura 29 - Vários tipos de módulos drivers. Da esquerda para a direita: buzzer, leds e motores DC.

Exercícios práticos

Controlar um motor DC com um transístor

Para controlarmos um motor DC com um transístor, teremos de criar o seguinte circuito em baixo, usando o transístor IRF520 (MOSFET – *enhancement mode n-channel*) e um díodo.

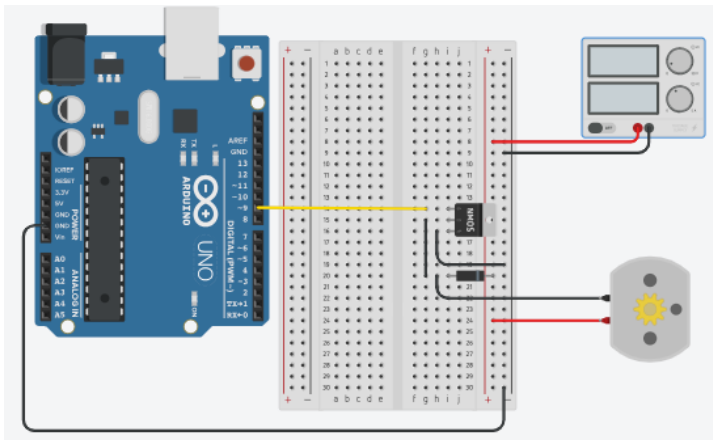


Figura 30 - Controlar um motor DC usando um transistor.

Quanto ao código, vamos escrever algo bastante simples que permita colocar o motor a rodar e a parar intercaladamente de 1 em 1 segundo.

```
int pinMotor = 9;

void setup() {
  pinMode(pinMotor, OUTPUT);
}

void loop() {
  digitalWrite(pinMotor, HIGH);
  delay(1000);
  digitalWrite(pinMotor, LOW);
  delay(1000);
}
```

Exercício

Partindo dos conhecimentos adquiridos até aqui, adapta o código para que o motor aumente e diminua a sua velocidade de rotação com um padrão orgânico semelhante à respiração de um ser humano.

Exercício

Utiliza o código e o circuito para controlar uma fita de leds, usando uma fonte de alimentação adequada.

Acender um candeeiro com um relé

Neste exercício vamos usar um relé para ligar e desligar uma lâmpada.

Para isso vamos usar o seguinte circuito:

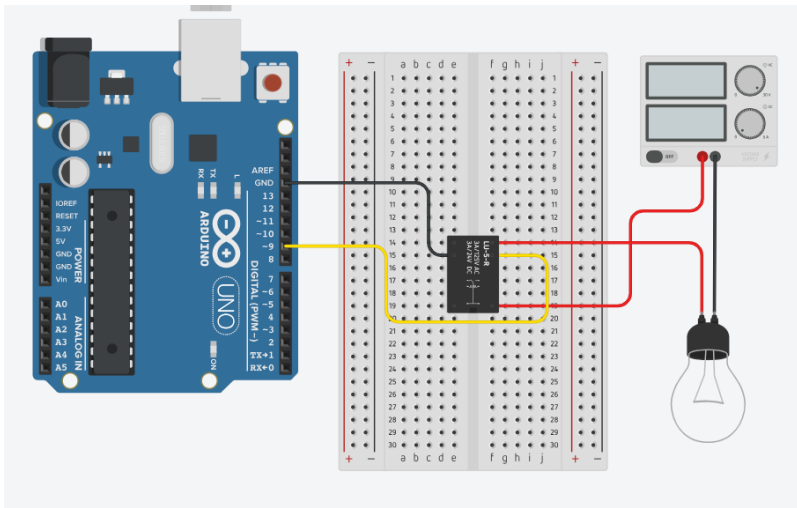


Figura 31 - Circuito para acender uma lâmpada com um relé.

O código que vamos utilizar tem como objetivo piscar a lâmpada cada 1 segundo.

```
int relayPin = 9;

void setup() {
  pinMode(relayPin, OUTPUT);
}

void loop() {
  digitalWrite(relayPin, HIGH);
  delay(1000);
  digitalWrite(relayPin, LOW);
  delay(1000);
}
```

Controlar um servo motor

Um *servo motor* é um tipo de motor bastante usado em robótica que consiste na combinação de um motor DC, um sistema de mudanças/velocidades, um sistema de controle e um sensor de posicionamento.

Esta combinação permite rotações precisas, com velocidades relativamente altas (mas inferiores a um motor DC) e alto torque. O tipo de *servo motor* mais comum tem um ângulo de rotação de apenas $\pm 180^\circ$, mas existem modelos (mais caros) que permitem uma rotação total.

Ao contrário dos motores DC, o *servo motor* tem três terminais: dois para a corrente e um para o controle. No geral, é mais caro que os motores DC.



Figura 32 - Servo motor (retirado de seedstudio.com)

Neste exercício vamos controlar um *servo motor* para rodar autonomamente de modo semelhante a uma escova limpa para-brisas. Para isso, vamos criar este simples circuito:

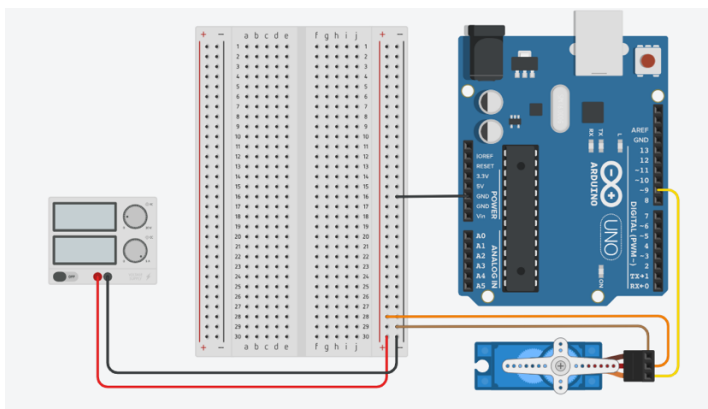


Figura 33 - Circuito para usar um servo motor.

O código apresenta algumas novidades.

A primeira linha serve para incluir uma biblioteca específica para controlar *servo motors* (`#include <Servo.h>`).

A segunda linha cria uma instância do novo objeto Servo, camado *myservo*.

A função `myservo.attach(9)` liga o recém criado objeto ao pino 9.

Por fim, a função `myservo.write(10)` indica ao servo o ângulo em que se deve posicionar.

```
#include <Servo.h>
Servo myservo;
int pos = 0;

void setup() {
    myservo.attach(9);
}

void loop() {
    myservo.write(10);
    delay(500);
    myservo.write(220);
    delay(500);
}
```

C8. Eletrónica digital: sensores

Sobre o capítulo

Objetivos de aprendizagem

- Ligar sensores analógico a um microcontrolador e obter dados coerentes com o estímulo físico correspondentes;
- Ligar sensores digitais a um microcontrolador e obter dados coerentes com o estímulo físico correspondentes;
- Controlar atuadores com os dados recolhidos dos sensores.

Conteúdo

- Tipos de sensores;
- Ligar sensores analógicos
- Ligar sensores digitais
- Monitorizar no monitor serial os dados recolhidos
- Relacionar dados recolhidos com ações dos atuadores

Recursos necessários

- Sensor de pressão (botão)
- Sensor de rotação (potenciómetro)
- Sensor de luminosidade (foto resistência)
- Sensor de toque (material condutor: tinta condutora, fio elétricos, etc.)

Tipos de sensores (uma pequena amostra)

Os sensores permitem obter informação sobre o mundo físico que nos rodeia, e por isso encontram-se presentes em inúmeros dispositivos que usamos: os sensores de estacionamento de um automóvel, o termostato de um aquecedor, etc.

Conforme o tipo de informação que transmitem, os sensores podem ser digitais ou analógicos, e podem ser binários ou apresentarem um intervalo de valores para o estímulo que “sentem”.

Segue-se uma pequena amostra de sensores e suas funções:

Botão

Um botão é um sensor que deteta pressão, geralmente de forma binária, ou seja, deteta se está pressionado ou não. A maioria dos botões são na verdade interruptores, que interrompem o circuito quando pressionados. Os botões podem ser NA (normalmente aberta) ou NF (normalmente fechada): um NA interrompe o circuito sempre que não é pressionado, e vice-versa. Podem ainda ser de travamento (ON-ON) – o seu estado permanece inalterado quando se retira o dedo - ou momentâneos – o seu estado retorna ao estado original quando se refira o dedo.

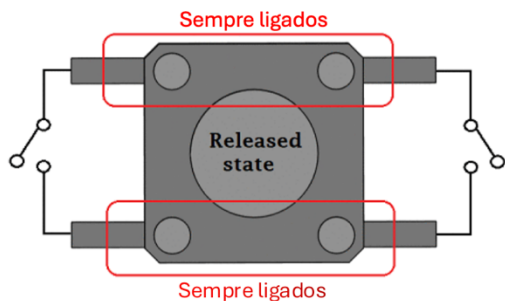


Figura 34 - À esquerda um botão NA para breadboard; à direita o esquema de ligação interno.

Para ligar um botão/interruptor liga-se geralmente aos pinos digitais do microcontrolador, uma vez que geram dados binários (ligado/desligado).

Para usar um botão num microcontrolador, teremos de usar uma estratégia específica para o tipo de botão: o NA usa um esquema *pull-down* em que o microcontrolador recebe 1 (HIGH) quando o circuito é fechado; um NF usa um esquema *pull-up* em que o microcontrolador recebe 0 (LOW) quando o circuito é aberto. Em ambos os casos, é necessário utilizar uma resistência, quer seja externa ou interna ao microcontrolador (se a tiver, temos de a ativar).

Potenciômetro

Os potenciômetros são resistências variáveis, que variam a intensidade da sua resistência de acordo com o rodar do manípulo. O valor que caracteriza o potenciômetro é dado em ohm Ω e corresponde à resistência máxima do potenciômetro.

Os potenciômetros têm geralmente 3 terminais, um para a V_{cc} (corrente), terra e resistência variável.

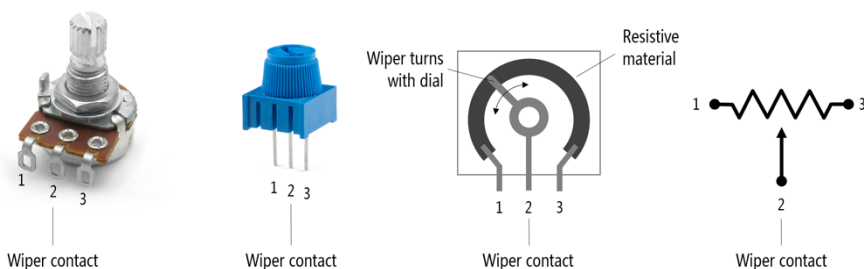


Figura 35 - da esquerda para a direita: potenciômetro de montagem, potenciômetro de prototipagem, diagrama funcional, esquema técnico.

Foto resistência

À semelhança dos potenciômetros, as foto resistências são igualmente resistências variáveis, mas a sua variação está dependente da quantidade de luz incidente na sua célula.

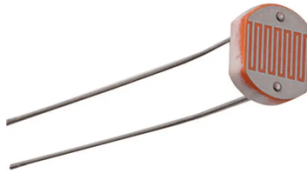


Figura 36 - foto resistência.

Outras estratégias e componentes que podem servir de sensores

Existem inúmeras formas de captar manifestações/alterações do/no nosso mundo físico. Enquanto que algumas dessas formas estão dependentes de equipamento dispendioso e complexo, outras formas utilizam componentes banais que encontramos nos objetos do dia-a-dia.

Um desses exemplos são as **ventoinhas** utilizada para refrigerar componentes eletrônicos dentro de um computador (fonte de alimentação, processador, etc.), as quais são baseadas num simples motor DC (sem escovas ou *brushless*), com algumas semelhanças àquele com que já trabalhámos. Um motor DC tem a valência de atuador, mas também de sensor, podendo ser utilizado para captar fenómenos físicos como o vento, se lhe acoplarmos uma hélice.



Figura 37 - ventoinha de computador 12V, sem escovas.

Outro bom exemplo é a exploração do conceito de **capacitância** para criar sensores que identifiquem quando um objeto condutor é tocado. A capacitância é um conceito complexo que pode, na sua essência, ser visto como a grandeza

Montar o esquema conforme a imagem, usando uma resistência de 10kΩ.

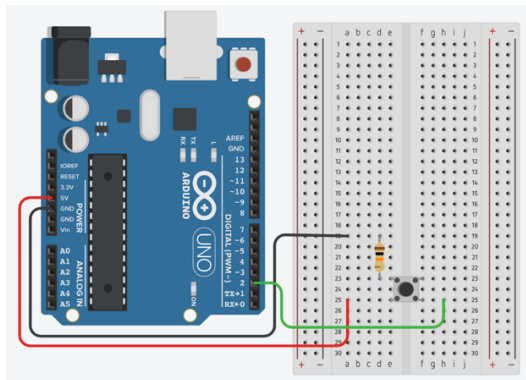


Figura 38 - Esquema de ligação de botão NF (com pull-down)

O código vai ler o valor de entrada do pino 2 e escrever essa informação na janela de monitorização serial.

```
int buttonPin = 2;
int buttonState = 0;

void setup() {
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  buttonState = digitalRead(buttonPin);
  Serial.println(buttonState);
  delay(10);
}
```

Exercício

Faz uma combinação deste exercício prático com o exercício prático da página 44, *LED externo a piscar*, modificando o circuito e o código para acender o LED com o pressionar do botão.

Ligar potenciômetro e ler valor em monitor serial

Montar o esquema conforme a imagem, usando um potenciômetro de 10kΩ.

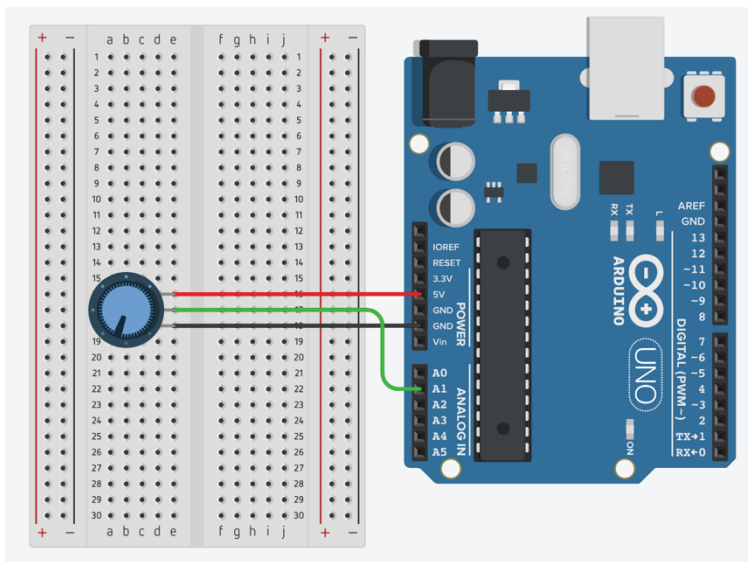


Figura 39 - Esquema de ligação de potenciômetro.

```
char potPin = A1;
int potState = 0;

void setup() {
    pinMode(potPin, INPUT);
    Serial.begin(9600);
}

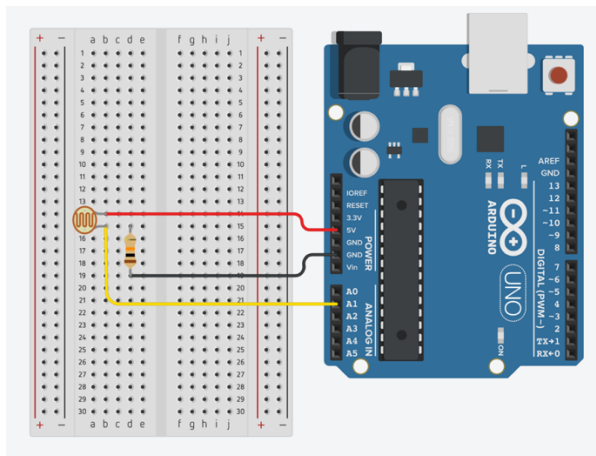
void loop() {
    potState = analogRead(potPin);
    Serial.println(potState);
    delay(10);
}
```

Exercício

Fazer uma combinação deste exercício prático com o exercício prático *Alterar a intensidade de um LED*, modificando o circuito e o código para alterar a intensidade do LED através do potenciômetro.

Ligar foto resistência e ler sinal no monitor serial

Fazer o seguinte circuito, com uma resistência de 10kΩ. Reutilizar o código do exercício anterior.



Exercício

Fazer uma combinação deste exercício prático com o exercício prático *O besouro a afinar*, modificando o circuito e o código para alterar a frequência do besouro através da luminosidade.

Sensor de vento usando uma ventoinha de computador

Neste exercício vamos usar uma ventoinha de computador (ou um motor DC) para captar informação relacionada com o vento. Para isso usamos um circuito bastante simples, onde ligamos a terra da ventoinha ao pino analógico A1.

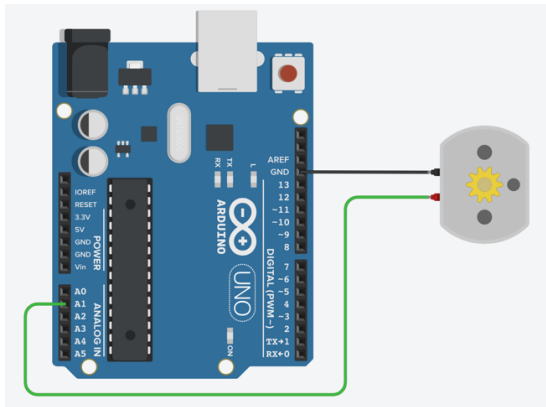


Figura 40 - ligação de motor DC / ventoinha de computador como sensor.

Quanto ao código, utilizamos o seguinte que é muito parecido ao do exercício *Ligar potenciómetro e ler valor em monitor serial*.

```
char fanPin = A1;
int fanState = 0;

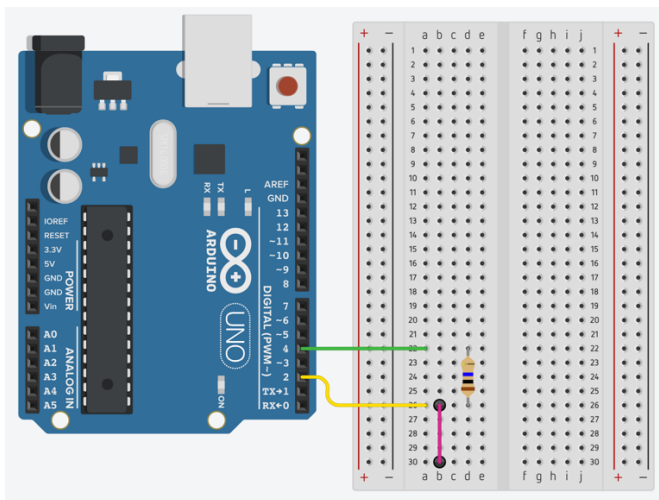
void setup() {
    pinMode(fanPin, INPUT);
    Serial.begin(9600);
}

void loop() {
    fanState = analogRead(fanPin);
    Serial.println(fanState);
    delay(10);
}
```

Sensor de toque através da capacitância

Neste exercício vamos criar o fenômeno da capacitância para fazer um sensor que consegue identificar quando e como um objeto condutor é tocado.

Vamos criar este simples circuito, usando uma resistência de 10M Ω (caso não se tenha acesso, ligar várias de menor valor em série). O fio cor-de-rosa é o fio que deverá ligar aos objetos que irão servir de sensor de toque (fio, chapa, etc.)



Para o código, teremos de descarregar e instalar a biblioteca <CapacitiveSensor>.

```
#include <CapacitiveSensor.h>

CapacitiveSensor cs_4_2 = CapacitiveSensor(4, 2);

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

void loop() {
  long toque = cs_4_2.capacitiveSensor(30);
  Serial.println(toque);
  delay(10);
}
```

Exercício

Fazer uma combinação deste exercício prático com o exercício prático *Alterar a intensidade de um LED*, modificando o circuito e o código para alterar a intensidade do LED através do toque no elemento condutor do circuito.

C9. Controlo de software com um microcontrolador

Sobre o capítulo

Objetivos de aprendizagem

- Enviar mensagens de controlo para um computador/software a partir de um MCU, usando o protocolo serial;
- Utilizar mensagens de controlo enviadas do MCU para alterar parâmetros num software de reprodução, geração e processamento de imagem e som (TouchDesigner);

Conteúdo

- Introdução ao TouchDesigner;
- Configurar o envio de mensagens entre Arduino e TouchDesigner;

Recursos necessários

- Computador com TouchDesigner e Arduino
- Microcontrolador
- Sensor (indiferenciado)

Software TouchDesigner – introdução e ligação ao MCU

O TouchDesigner é um software particularmente vocacionado para a criação de obras de videoarte interativas, generativas, não lineares e/ou imersivas em tempo-real. É baseado num modelo de programação por *nodes*, onde o artista cria uma teia de processos sobre os parâmetros de som, de imagem e de interatividade.

A comunicação entre o microcontrolador e o TouchDesigner pode ser bidirecional e baseada em diversos protocolos (OSC, série, etc.). Neste manual vamos abordar o envio de mensagens do microcontrolador para o TouchDesigner usando o protocolo de comunicação serial.

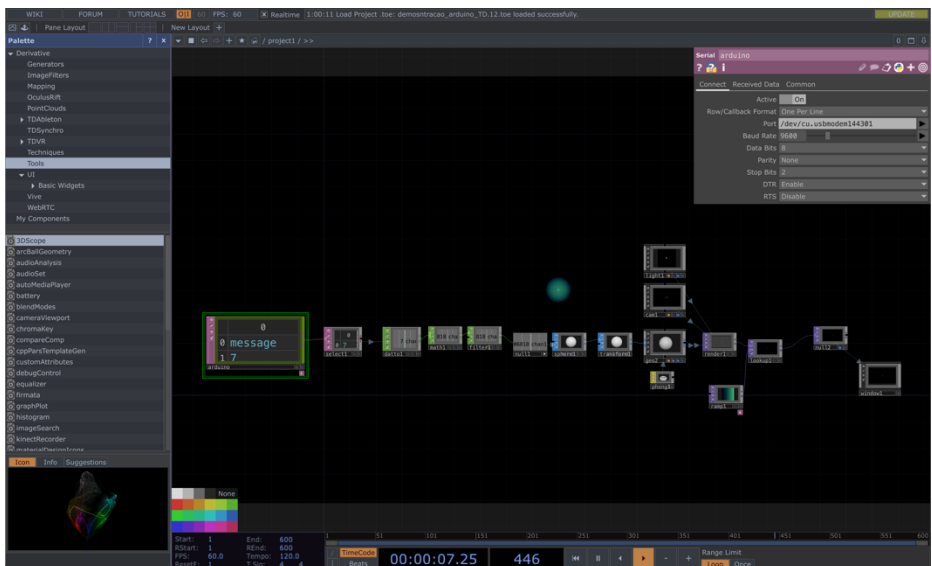


Figura 41 - Interface gráfico do utilizador para o TouchDesigner

Ligação serial a partir do Arduino IDE para o TouchDesigner

É bastante simples fazer o envio de mensagens do Arduino (ou outro MCU) para o TouchDesigner através do protocolo serial, basta enviá-las através da função `Serial.print()/Serial.println()`. Por exemplo, se pretendermos enviar três variáveis (*pressão, distância e rotação*) escreveríamos o seguinte pedaço de código, algures para no final do `void loop{}`:

```
void loop() {
    Serial.print(pressao);
    Serial.print(" ");
    Serial.print(distancia);
    Serial.print(" ");
    Serial.println(rotacao);
}
```

Para receber essa informação no TouchDesigner, é necessário adicionar o operador “DAT > Serial” a partir da janela de operadores.

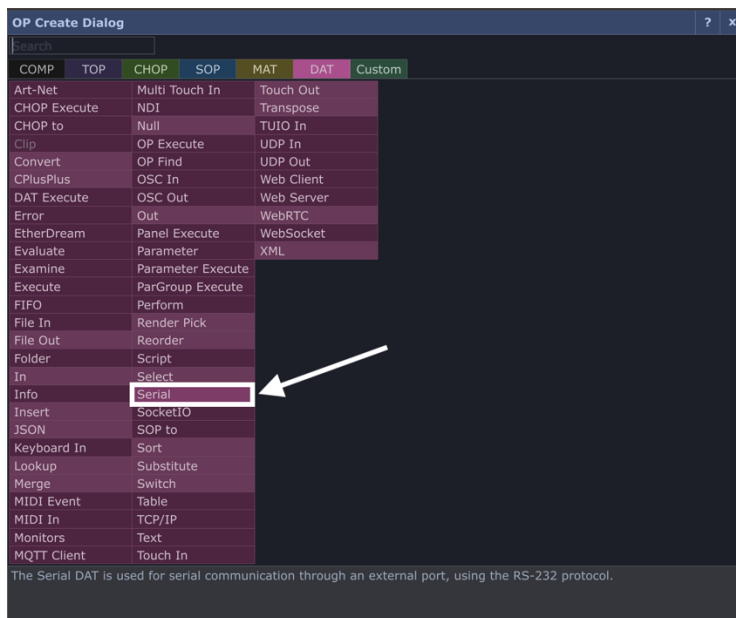


Figura 42 - Janela de operadores do TouchDesigner

Depois de colocado o operador na janela de projeto do TouchDesigner, é necessário configurar os parâmetros do operador, tendo em consideração – principalmente – os parâmetros da Baud Rate (que deve ser igual ao sketch de Arduino) e a porta (deve ser a USB). Depois de tudo configurado é necessário

ativar o operador, colocando “on” no parâmetro Active, mas atenção, para esta porta funcionar, é necessário que a porta Serial no Arduino IDE não esteja ligada (em caso de dúvida, fechar o Arduino IDE).

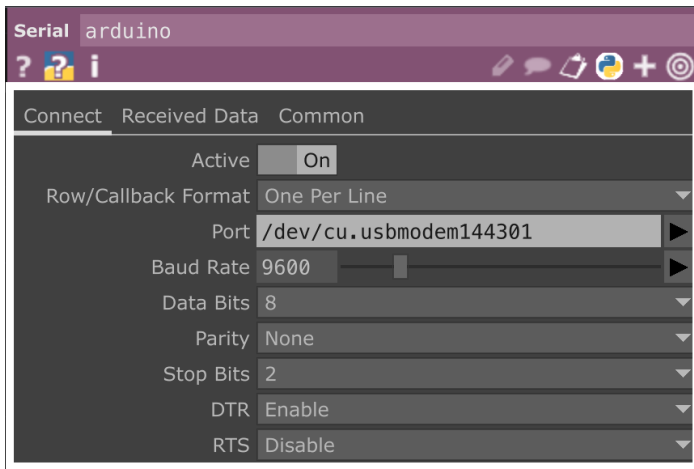


Figura 43 - Janela de configuração do operador Serial

Depois de configurada a ligação, as mensagens deverão começar a ser visíveis no operador Serial, procedendo-se de seguida a um conjunto de manipulações para adaptar as mensagens às necessidades criativas pretendidas.

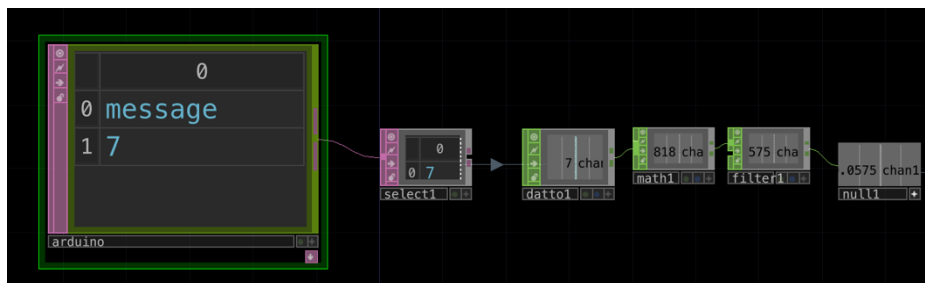


Figura 44 - Operador Serial (à esquerda), seguido de outros operadores para manipulação das mensagens recebidas.

