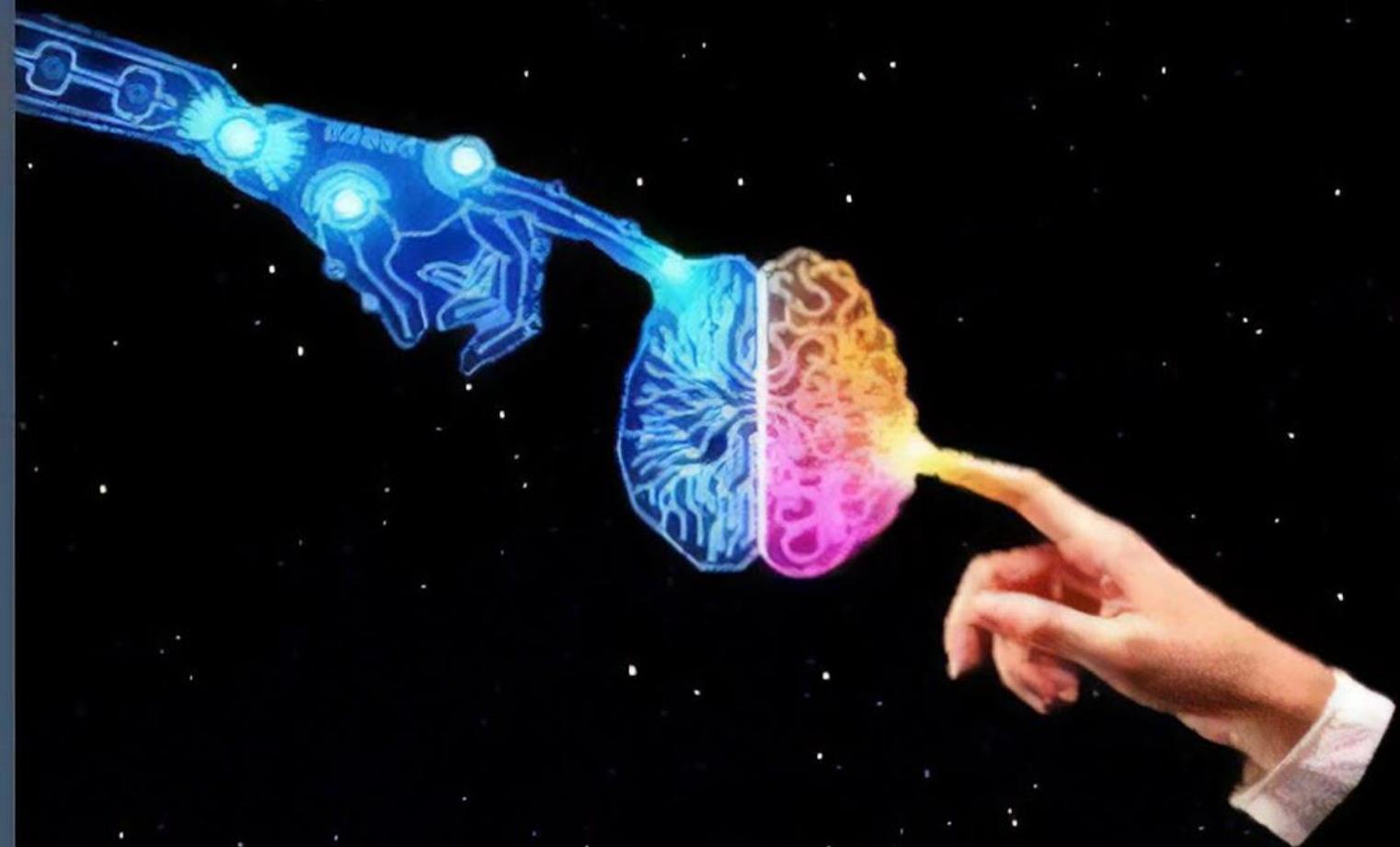


INSTITUTO FEDERAT DE SÃO
PAULO: CÂMPUS JACAREÍ



ROBÓTICA COMPUTACIONAL

SUMÁRIO

INTRODUÇÃO.....	6
HISTÓRICO.....	7
APLICAÇÕES DA ROBÓTICA.....	10
Aplicação Industrial	10
Aplicação na Medicina	12
Aplicação na Limpeza	15
Aplicação no Transporte	16
Aplicação Espacial	18
Aplicação na Agricultura e Pecuária	20
CONCEITOS BÁSICOS.....	22
Introdução ao Arduíno.....	22
O que é Arduíno	22
Hardware do Arduino	22
Portas Analógicas e Digitais.....	23
Instalação e Explicação do Software.....	24
Instalação	24
Explicação IDE	25
Monitor Serial	27
Biblioteca	29
Conceitos básicos de eletrônica	30
Resistores e Lei de Ohm.....	30
Código de Cores	30
Entrada e Saída.....	32
Multímetro.....	32
Continuidade.....	33
Corrente Elétrica	33
Tensão de Corrente Alternada	34
Tensão de Corrente Contínua	34
Resistência	34
TINKERCAD	35
Projetos 3D (1)	35
Circuits (2).....	37

Blocos de Código (3)	42
Lições (4)	45
Criar Projeto (5)	45
INTRODUÇÃO A LINGUAGEM C PARA ARDUINO	46
Estruturas	46
Estruturas de Controle.....	46
If / else	46
While.....	47
Do-While	47
For	48
Break.....	48
Continue.....	49
Return	49
Elementos de Sintaxe	49
Operadores	50
Aritméticos.....	50
Comparação	50
Booleanos.....	51
Composição.....	51
Valores.....	51
Constantes	51
Tipos de Dados	52
Funções.....	53
Entrada e Saída Digital	53
Entrada e Saída Analógica	53
Temporização.....	53
Funções matemáticas	54
Bits e bytes.....	54
GUIA DA CAIXA DE ROBÓTICA.....	55
Fonte 9v Arduino Bivolt e Cabo USB.....	55
Protoboard.....	55
Jumpers	56
Resistor	57
Diodo	57
Capacitor Cerâmico	57
Capacitor Eletrolítico	58

LED	58
Chave Táctil (Push-Button).....	61
Potenciômetro	65
Sensor de Luz LDR	68
Sensor de Temperatura NTC e PTC.....	70
Buzzer.....	72
Display LCD	74
Sensor de Distância Ultrassônico	80
Sensor Óptico Reflexivo	84
Servo Motor	87
Motor de Passo	90
Módulo Relé 1 Canal.....	93
OUTROS SENSORES E ELEMENTOS DA ROBÓTICA.....	95
LED RGB	95
Sensor NTC 10K.....	99
Sensor RGB	101
Keypad.....	106
Display 7 Segmentos.....	109
EXPERIMENTOS AVANÇADOS.....	117
Semáforo comum.....	117
Semáforo interativo.....	118
Sequencial de LEDs.....	119
LED RGB controlado por botões.....	120
LED RGB controlado por botões e potenciômetro	121
Criando cores com LED RGB utilizando Monitor Serial.....	122
Sensor de luz LDR com saída em LEDs	123
Sensor de luz LDR com sinalizador de LEDs e Buzzer.....	124
Jogo das Palavras Embaralhadas	125
Alarme de temperatura com Buzzer	126
Piano com buzzer	127
Sensor Ultrassônico com Buzzer e Led	128
Relógio com hora e data	129
Sensor de distância com LEDs	130
Sensor de distância e motor	131
Sensor óptico reflexivo com botão de reset	132
Jogo da Tabuada.....	133

Servo motor com Sensor LDR.....	134
Movendo o Servo motor com botões.....	135
Acelerando e desacelerando o motor com luz.....	136
LEDs e Sensor RGB.....	137
Fechadura eletrônica controlada por senha	138
Keypad e Display 7 segmentos.....	139
Semáforo Interativo com Display 7 segmentos.....	140
Sensor de Umidade e Display LCD.....	141
DHT11 com Buzzer e Led	142
DHT22 com Sensor LDR e Sensor Ultrassônico	143
Sensor DHT22 com alarme RGB	144
Dado eletrônico.....	145
Monitorando Luminosidade e Temperatura com LCD.....	146
Jogo da memória - Genius	147
PROJETOS.....	148
Cubo de Led	148
Controle de Acesso usando Leitor RFID.....	149
Árvore de Natal	150
Alarme inteligente.....	151
Sistema de Irrigação	152
Dispenser automático de álcool em gel.....	153
Robô seguidor de linha e desvia de obstáculos.....	154
Controlando Projeto com Controle Remoto	155
Acender lâmpada com palmas	156
Separando objetos por cor com braço mecânico.....	157
Painel de Senhas de Atendimento.....	158
BIBLIOGRAFIA	159

INTRODUÇÃO

Robótica é definida como o estudo dos robôs e da sua capacidade de sentir e agir no mundo físico de forma autônoma e intencional. (MATARIC, 2014). A Robótica Educacional tem como objetivo conhecer os diversos tipos de sensores, elementos e programação, no caso C para Arduino, além de promover estudo interdisciplinar com matérias como física, matemática e biologia, aprimorando o raciocínio lógico.

Mas o que seria um robô? Um robô é um manipulador multifuncional reprogramável projetado para mover materiais, peças, ferramentas ou dispositivos especializados, através de movimentos programados para a execução de uma variedade de tarefas. (RIA, 1979 *apud* PARENT; LAURGEAU, 1984, p. 11)

Essa apostila tem como objetivo auxiliar os alunos na disciplina de Robótica Computacional do Curso Integrado em Técnico de Informática do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus Jacareí, abordando os assuntos apresentados durante o ano, como Arduino, aplicações, conceitos, elementos da robótica, sensores, simulador Tinkercad, linguagem C para Arduino e práticas de exercícios.

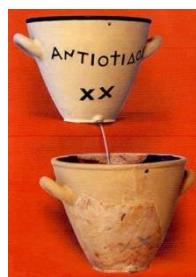
Criada em 2020 pela aluna Rauany Martinez Secci, com o auxílio dos professores Ana Paula Abrantes de Castro e Shiguemori e Tardelli Ronan Coelho Stekel, por meio do projeto de ensino voltado a suporte aos alunos de Robótica.

HISTÓRICO

No início da história da robótica, o conceito de robô era relacionado a mitos, onde mecanismos ganhavam vida; um dos maiores sonhos do ser humano, era criar uma máquina que tivesse a função de um servo, capaz de pensar e agir, satisfazendo o desejo de seu senhor.

Ctesibius foi um matemático e engenheiro grego que viveu cerca de 285-222 a.C. em Alexandria. Acredita-se que ele foi a primeira pessoa a construir um dispositivo robótico, um aparelho para medir o tempo: clepsidra (relógio de água)

Figura 1: Clepsidra.



Fonte: Imagem tirada de (Água Rara, 2015).

Heron de Alexandria foi geômetra, matemático e engenheiro grego (10 d.C. - 80 d.C.). Construiu diversos sistemas robóticos, como a primeira máquina de vender bebidas e o primeiro motor a vapor.

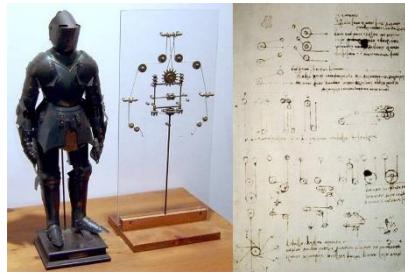
Figura 2: Sistemas Robóticos Heron de Alexandria: (a) primeira máquina de vender bebida e (b) primeiro motor a vapor.



Fonte: (a) Imagem tirada de (World Observer by Claudia & Vicente, 2013) e (b) Imagem tirada de (Wikipedia, 2012).

Leonardo da Vinci foi cientista, matemático, engenheiro, inventor, anatomista, pintor, escultor, arquiteto, botânico, poeta e músico (Anchiano, 15 de abril de 1452 - Amboise, 2 de maio de 1519). Teve ideias muito além do seu período, como um helicóptero, tanque de guerra, o uso da energia solar e a calculadora, mas o que passou de uma ideia e foi desenvolvido foi o “Robô de Leonardo”, que consistia em um cavaleiro que se movia sozinho, como se houvesse uma pessoa em seu interior.

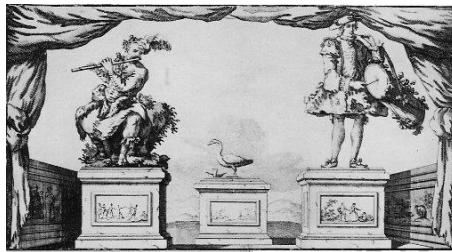
Figura 3: Robô de Leonardo.



Fonte: Imagem tirada de (Thizer, 2018).

Jacques de Vaucanson foi inventor e artista francês (Grenoble, 24 de fevereiro de 1709 — 21 de novembro de 1782). Criou o primeiro robô funcional, um robô que tocava flauta e outro pandeiro e um pato mecânico. Essas novas invenções deram espaço para criações de teorias literárias, que diziam sobre o medo dos seres humanos, de serem substituídos por suas próprias criações.

Figura 4: Robô Flautista, Pato mecânico, Robô Pandeirista.

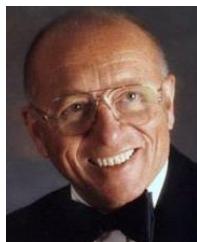


Fonte: Imagem tirada de (Wikiwand, 2004).

Assim, Isaac Asimov, em 1940 lança diversas publicações voltadas a Robótica, primeira vez que foi usado o termo “robótica”, afirmando que os robôs surgiram para ajudar os homens em suas tarefas e não para substituí-los. A partir disso ele escreveu “as três leis da robótica”: primeira lei: “Um robô não pode ferir um ser humano ou, por falta de ação, deixar que um ser humano se fere.”, segunda lei: “O robô deve obedecer às ordens dadas pelos seres humanos, exceto se tais ordens estiverem em contradição com a primeira lei” e a terceira lei: “Um robô deve proteger sua existência na medida em que essa proteção não estiver em contradição com a primeira e a segunda lei”.

Na década de 1950, foram fabricados os primeiros robôs para a indústria, o primeiro a construir tal robô (Unimate) foi Joseph F. Engelberger, engenheiro e empresário considerado o “pai da robótica”, que foi vendido para General Motors, com a função de trabalhar na linha de montagem.

Figura 5: (a) Joseph F. Engelberger “pai da robótica” e (b) primeiro robô industrial-Unimate.



(a)



(b)

Fonte: (a) Imagem tirada de (Wikipedia, 2015) e (b) Imagem tirada de (Unimate, 2020).

A partir disso, a robótica entrou em contínua expansão, e até os dias de hoje podemos observar como essa tecnologia tem se inovado, com diferentes finalidades, sendo utilizado em indústrias, na ciência, pesquisa, entre outras funções.

Figura 6: Sophia (robô), o primeiro robô a ganhar cidadania no mundo.



Fonte: Imagem tirada de (Wikipedia, 2017).

APlicações da robótica

Nos dias de hoje, a robótica está presente em muitas das tarefas da vida cotidiana. Os robôs surgiram primeiramente voltado à indústria, mas estão em nossas vidas domésticas, na pesquisa, saúde, entre outros.

Aplicação Industrial

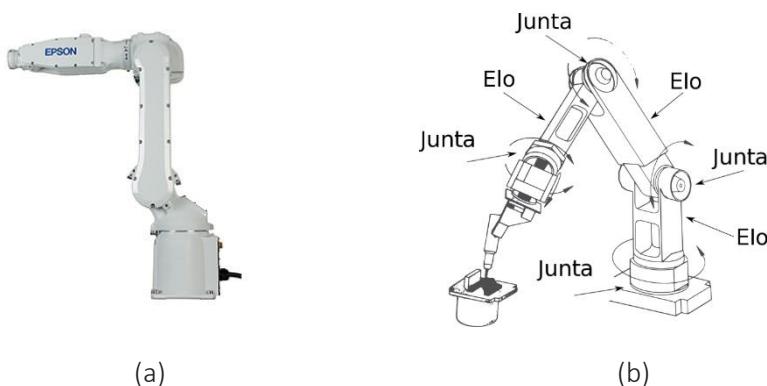
A robótica industrial é o estudo de robôs utilizados em instalações industriais, como em linhas de montagem de automóveis ou engarrafamento de bebidas. (VÁZQUEZ, 2015).

Inicialmente, os robôs vieram para substituir ações humanas repetitivas dentro de uma linha de produção, trazendo diversos benefícios para a indústria como: flexibilidade, alta produtividade, melhor qualidade dos produtos e aumento da qualidade da vida humana, pelo desempenho de tarefas indesejadas.

Nesta aplicação incluem robôs articulados, robôs SCARA, robôs cartesianos (robôs X, Y e Z), robôs cilíndricos, robô polar e robô delta. Sua programação varia de cada função, alguns são programados para repetirem a mesma ação por indefinidas vezes com grande precisão (CARRARA, 2015).

Os **robôs articulados** são os mais utilizados na indústria, podem ser chamados também de robô quatro eixos ou robô seis eixos, varia por conta da quantidade de juntas que ele tem. Suas **vantagens** são: fácil de alinhar as coordenadas, flexibilidade, pequeno espaço utilizado e alta velocidade. Suas **desvantagens** são: programação complicada. Os encontramos nas **funções** de manipulação de vidro, soldagem, corte de aço, embalagem de alimentos e paletização.

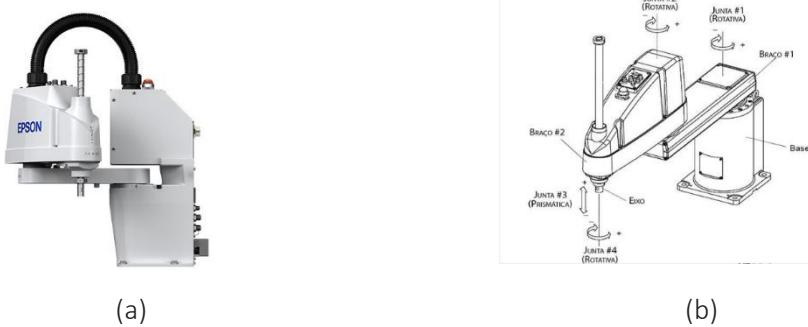
Figura 7: Robô Articulado: (a) robô articulado de seis eixos e (b) estrutura do robô articulado.



Fonte: (a) Imagem tirada de (Epson, 2020) e (b) Imagem tirada de (Laboratório de eletrônica, 2016).

Os **robôs SCARA** são utilizados em montagem, especializados em movimentos laterais, composto por duas juntas paralelas, que dão suporte para o plano onde estiverem. Suas **vantagens** são: fácil de programar, boa repetibilidade e alta velocidade. Suas **desvantagens** são: só pode ser utilizado em superfícies planas. Os encontramos nas **funções** de montagem, aplicação biomédica, embalagens, inspeção e aplicação de cola.

Figura 8: Robô SCARA: (a) robô SCARA e (b) estrutura do robô SCARA.



Fonte: (a) Imagem tirada de (Epson, 2020) e (b) Imagem tirada de (Researchgate, 2018).

Os **robôs cartesianos** possuem um formato retangular e três juntas prismáticas, possibilitando o movimento circular, também chamados de robôs retilíneos. Suas **vantagens** são: baixo custo, suporta cargas pesadas, operação simples e precisão de posição. Suas **desvantagens** são: movimento limitado, necessita de uma grande área para instalação e montagem complexa. Os encontramos nas **funções** de montagem, aplicações adesivas, carregar e descarregar materiais.

Figura 9: Robô Cartesiano: (a) robô cartesiano e (b) estrutura do robô cartesiano.



Fonte: (a) Imagem tirada de (Alibaba, 2020) e (b) Imagem tirada de (Controle e Automação industrial 3, 2014).

Os **robôs cilíndricos** têm uma ou mais juntas rotativas e prismática para a conexão. Suas **vantagens** são: pequeno espaço utilizado, transporta grandes cargas e operação simples. Suas **desvantagens** são: não é muito utilizado, não tem grande êxito no movimento rotativo e não conseguem pular obstáculos. Os encontramos nas **funções** de fundição, aplicação de revestimento e transporte de painéis.

Figura 10: Robô Cilíndrico: (a) robô cilíndrico e (b) estrutura do robô cilíndrico.



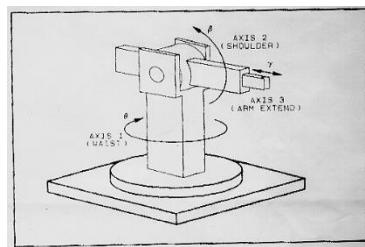
Fonte: (a) Imagem tirada de (Citisystems, 2016) e (b) Imagem tirada de (Blog do Professor Carlão, 2009).

Os **robôs polares** possuem um eixo de articulação e um braço giratório; suas juntas servem para conectar a base e o braço, juntamente com duas juntas rotativas e uma junta linear. Suas **vantagens** são: alcançar tudo ao redor, requer pequeno espaço e grande volume de trabalho. Suas **desvantagens** são: não alcançar acima de si e sistema sofisticado de controle. Os encontramos nas **funções** de empilhamento, moldagem, fundição e manipulação de vidros.

Figura 11: Robô Polar: (a) robô polar e (b) estrutura do robô polar.



(a)



(b)

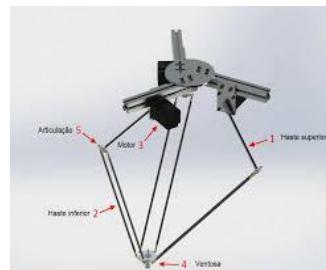
Fonte: (a) Imagem tirada de (Citisystems, 2016) e (b) Imagem tirada de (Researchgate, 2019).

Os **robôs delta** possuem ligações paralelas conectadas a base, semelhante a uma aranha, podem ser chamados também de robôs de link paralelo. Suas **vantagens** são: alta precisão e velocidade. Suas **desvantagens** são: operação complicada e controlador complexo. Os encontramos nas **funções** de simuladores de voo, automóveis e indústrias.

Figura 12: Robô Delta: (a) robô delta e (b) estrutura do robô delta.



(a)



(b)

Fonte: (a) Imagem tirada de (Mectrol, 2020) e (b) Imagem tirada de (Reposito roca, 2012).

Aplicação na Medicina

O uso da robótica voltado a medicina, apesar de ser uma aplicação recente, vem crescendo depressa. Este crescimento ocorre por conta das inúmeras vantagens que existem, como precisão, repetição, velocidade, confiabilidade e boa relação de custo/benefício (MIOTEC, 2018).

Mas isso não quer dizer que os médicos serão substituídos por robôs, principalmente em cirurgias robóticas, onde seus movimentos não são autônomos e sim guiados por um profissional capacitado. A função do robô é evitar erros humanos, que no caso, poderia ser um tremor na mão do cirurgião.

Na medicina, o robô é encontrado de diversas formas, sendo membros e órgãos artificiais, Telecirurgia, apoio a idosos e deficientes, cadeira de rodas, entre outros.

Os **robôs cadeira de rodas**, tem características específicas para serem consideradas robôs. Pode-se citar a fabricada pela empresa Freedom, chamada Millenium "C", que possui um banco automático, e a fabricada pela Toyota, i-Real, que possibilita subir e descer desníveis (NETO, 2013).

Figura 13: Cadeira de Rodas Robô: (a) Millenium C e (b) i-Real.



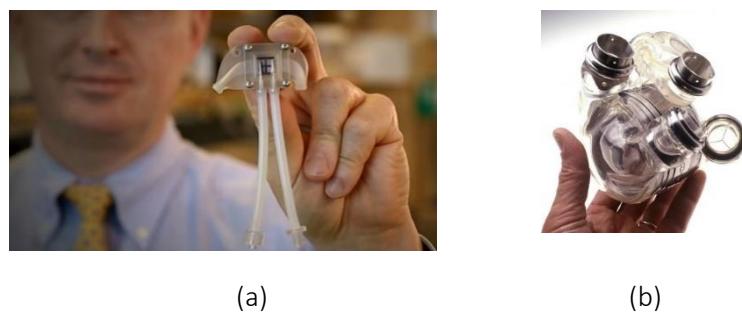
(a)

(b)

Fonte: (a) Imagem tirada de (Freedom, 2020), (b) Imagem tirada de (car body design, 2007), (c) Imagem tirada de (UFRN, 2013).

Os **robôs órgãos artificiais**, podem ser os membros das próteses ou funcionar de forma automatizada, como coração, estômago, rim, fígado e pulmão.

Figura 14: Órgãos artificiais: (a) rim artificial e (b) coração artificial.



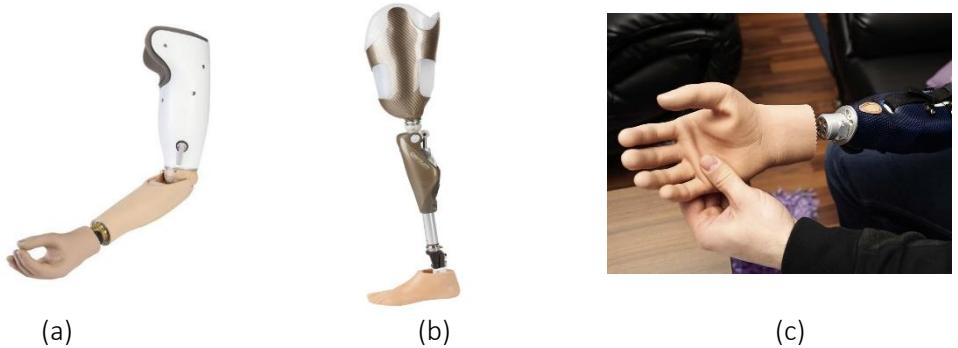
(a)

(b)

Fonte: (a) Imagem tirada de (SIMI, 2018) e (b) Imagem tirada de (Folha de São Paulo, 2017).

Os **robôs membros artificiais**, são vistos como um apoio ao deficiente, contém sensores de forma automatizada e possui um sistema de realimentação, por conta de o novo membro substituir os músculos do deficiente, que obedece aos comandos dados.

Figura 15: Membros artificiais: (a) prótese de braço; (b) prótese de perna e (c) prótese de mão.



Fonte: (a) Imagem tirada de (Choll Ortopedia, 2020), (b) Imagem tirada de (Ottobock, 2019) e (c) Imagem tirada de (Veja, 2015).

Os **robôs de apoio nos hospitais** têm serventia de auxiliar os enfermeiros, faz com que eles só tenham que tratar da saúde dos pacientes, sem se preocupar com refeições, leva de medicamentos ou roupa suja, pois esse será o papel do robô.

Figura 16: Robô Helpmade da Pyxis Corp.



Fonte: Imagem tirada de (Robôs na medicina, 2013).

Os **robôs cirurgiões** foram criados para reparar os erros humanos, atualmente eles são utilizados em operações delicadas que precisam de exatidão, sem o tremor humano, fazendo mínimo processo invasivo no paciente.

Os dois principais robôs cirurgiões são da Vinci, que tem as dimensões de um lápis, mas tem grande serventia em uma cirurgia por conta dos elementos que o robô tem, como uma câmera miniatura que auxilia o médico, e o AESOP (Automated Endoscopic System for Optimal Positioning), que responde a ordem por comando de voz do médico e leva um endoscópio que traz firmeza e estabilidade ao cirurgião.

Figura 17: Robôs Cirurgiões: (a) robô da Vinci e (b) robô AESOP.



Fonte: (a) Imagem tirada de (Saúde Business, 2019) e (b) Imagem tirada de (Robôs na Medicina UBI, 2013).

Outra ferramenta que a robótica agregou a medicina foi a **Telemedicina**, que é a capacidade de operar um robô cirurgião remotamente ou vários médicos participarem de uma mesma cirurgia estando em locais diferentes, sem precisar se deslocar até o local da cirurgia. O RP6 foi desenvolvido nesta ideia de Telemedicina, o médico faz consultas ou visitas médicas por meio dos robôs RP6.

Figura 18: Médico Robô Virtual RP6.



Fonte: Imagem tirada de (Robôs na Medicina UBI, 2013).

Aplicação na Limpeza

A limpeza de casas é uma tarefa que todos realizam, por esse motivo foram automatizados meios de limpeza, com robôs cortadores de grama, limpadores de piscinas, que varrem e aspiram. Também foram criados robôs para a limpeza dos oceanos.

Os **robôs aspiradores de pó** são autônomos e memorizam as regiões já passadas e passando por onde ainda falta ser aspirado, param e se recarregam sozinhos e quando carregados retornam de onde pararam. Alguns como o robô Romy (Figura 19(b)) possuem escovas para a limpeza de partes exteriores da casa e respondem a comandos de voz.

Figura 19: Robôs aspiradores de pó: (a) Robô Roomba 960 iRobot e (b) Romy.



Fonte: (a) Imagem tirada de (Irobot, 2020) e (b) Imagem tirada de (DocPlayer, 2016).

Os **robôs cortadores de grama**, tem um formato semelhante aos robôs aspiradores de pó. São capazes de se carregarem sozinhos e assim retornar ao serviço de poda.

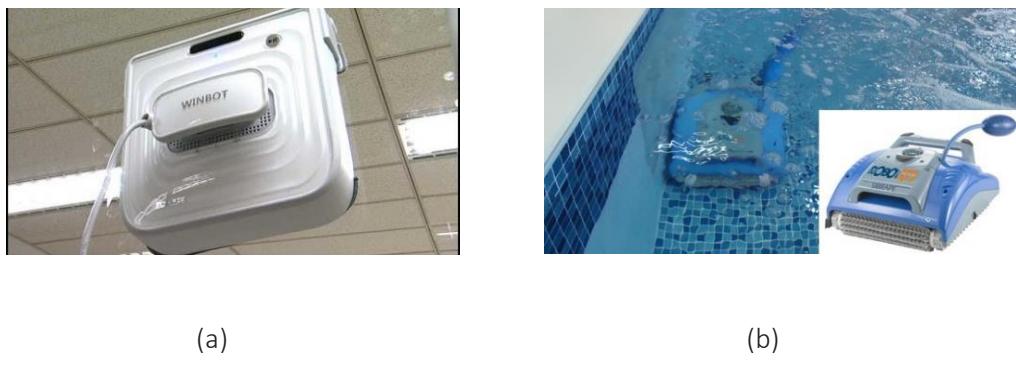
Figura 20: Automower robô cortador de grama.



Fonte: Imagem tirada de (Olhar digital, 2018).

Os **robôs limpadores de vidro** além de aspirar o pó das janelas eles desengorduram os vidros, já os **robôs limpadores de piscina** não apenas aspiram a sujeira como esfregam todo o chão da piscina com suas escovas.

Figura 21: Robôs limpadores: (a) limpador de vidro e (b) limpador de piscina.



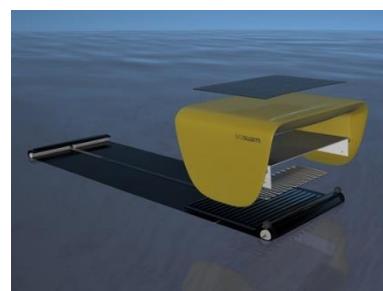
(a)

(b)

Fonte: (a) Imagem tirada de (YouTube, 2014) e (b) Imagem tirada de: (YouTube, 2013).

Outro tipo de limpeza com robôs que pode ser citado é de **superfícies marinhas**. O robô criado pelo MIT, chamado Sea Swarm, tem função de recolher petróleo e poluentes, evitando os danos causados por navios petrolíferos.

Figura 22: robô Sea Swarm.



Fonte: Imagem tirada de (Veja, 2010).

Aplicação no Transporte

Diversos meios de transporte já fazem o uso de aplicações robóticas, estes são os pilotos automáticos, mas precisam de supervisão. Os estudos e tentativas são de originar automóveis que sejam autônomos, exemplo disso é o carro que está sendo desenvolvido pela Google.

Figura 23: Google Car.



Fonte: Imagem tirada de (Business Insider, 2018).

Outros automóveis que estão tendo influência da robótica, são os de **transportes de carga**, que tem como função a eficiência na entrega de mercadorias. A empresa Amazon atualmente faz uso desses robôs para o transporte de mercadorias em seus centros de distribuição e está fazendo testes de robôs entregadores, chamados Amazon Scout.

Figura 24: Amazon Scout.



Fonte: Imagem tirada de (Mobile time, 2019).

Transportes utilizados em busca e salvamento também estão sendo automatizados, os chamados UAV - Unmanned Aerial Vehicles (veículos aéreos não tripulados), que são aeronaves e helicópteros sem pilotos, e AUV - Autonomous Underwater Vehicles (veículos submarinos autônomos), que são submarinos sem tripulação, para a busca de pessoas necessitadas.

Figura 25: Transportes utilizados em busca: (a) UAV (veículos aéreos não tripulados) e (b) AUV (veículos submarinos autônomos).



(a)



(b)

Fonte: (a) Imagem tirada de (Wikipedia, 2014) e (b) Imagem tirada de (MBARI, 2012).

Transportes de guerras, também estão sendo automatizados. Os robôs militares (tanques de guerra) têm uma tecnologia semelhante aos robôs de busca. Da mesma forma são utilizados os robôs Cypher, que voam como um helicóptero utilizado para espionagem de território inimigo.

Figura 26: Transportes de Guerra: (a) robôs tanque de guerra e (b) Cypher.



(a)



(b)

Fonte: (a) Imagem tirada de (Robôs não Industriais UBI, 2013) e (b) Imagem tirada de (Robôs não Industriais UBI, 2013).

Aplicação Espacial

Há quase 70 anos, os robôs fazem parte das explorações espaciais, incluindo as pesquisas sobre Marte, em relação a possibilidade de existência de vida, realizadas por robôs que vão até o planeta vermelho. Foram enfrentados inúmeros obstáculos para a ida ao espaço, principalmente com a formação do robô, que necessitaria de um material que fosse resistente à radiação e ventos solares, chegando ao seu destino como saiu do seu ponto de partida. Existem dois tipos de robôs espaciais: o *Remotely Operated Vehicle* (ROV) e o *Remotely Manipulator System* (RMS).

O **ROV** opera como um robô autônomo, que é levado até seu destino por meio de espaçonaves aterrissadoras. Suas características são rodas com movimentação e tração independentes além de braços e instrumentos robóticos. Exemplo desse é o robô Curiosity, citado abaixo.

O **RMS** é um braço mecânico, com função de imitar as funcionalidades de um braço humano, na aplicação espacial seu papel é auxiliar os astronautas a consertarem estações espaciais e satélites.

Pode ser citado os seis robôs que já saíram da Terra, auxiliando em pesquisas espaciais, são eles: Sojourner, Spirit, Opportunity, Curiosity, Lunokhod 1 e Yutu.

Sojourner, foi o primeiro veículo a circular em Marte, feito pela NASA, fazendo parte da missão Mars Pathfinder em 1997. Seu objetivo era determinar a composição das rochas do solo marciano, foi encerrado no dia 27 de setembro de 1997 quando a comunicação com a Terra foi perdida.

Figura 27: Sojourner.



Fonte: Imagem tirada de (Construção de um Rover Robô, 2015).

Spirit, nome escolhido em uma competição estudantil feita pela NASA, foi desenvolvido pela NASA para explorar Marte, lançado em junho de 2003 junto ao foguete Delta II, chegando em janeiro de 2004 na Cratera de Gusev, realizou suas tarefas até dia 22 de março de 2010.

Figura 28: Spirit.



Fonte: Imagem tirada de (Plano news, 2018).

Opportunity, pousou no Meridiani Planum depois de três semanas que Sprit aterrissou, no dia 25 de janeiro de 2004, buscava sinal de água no solo do planeta vermelho, realizou suas

tarefas até junho de 2018, quando houve uma tempestade de poeira fazendo com que drenasse a bateria do robô.

Figura 29: Opportunity.



Fonte: Imagem tirada de (Time, 2019).

Curiosity foi lançado em 26 de novembro de 2011, chegando no dia 6 de agosto de 2012. Seus objetivos são: estudar o clima de Marte, estudo da água no local, presença de compostos orgânicos e habitabilidade planetária para a exploração humana. Atualmente ele ainda está em funcionamento (ano de 2020).

Figura 30: Curiosity.



Fonte: Imagem tirada de (NASA, 2011).

Lunokhod 1, traduzido aquele que anda na lua, foi o robô enviado para a Lua pela União Soviética, o lançamento aconteceu em 10 de novembro de 1970, chegando em 7 de dezembro de 1970. Foi criado para medir a radiação da superfície lunar e estudar sua superfície. A missão terminou em setembro de 1971.

Figura 31: Lunokhod 1.



Fonte: Imagem tirada de (Wikipedia, 2013).

Yutu, nome que faz referência ao coelho de estimação de Chang'e, foi criado pela China para explorar a Lua na missão Chang'e 3, lançado em 1 de dezembro de 2013 e pousando em 14 de dezembro de 2013. Estudou a geologia da Lua para futuras missões espaciais, ficou até o dia 31 de julho de 2016.

Figura 32: Yutu.



Fonte: Imagem tirada de (Wikipedia, 2013).

Aplicação na Agricultura e Pecuária

Os serviços do campo são pesados e repetitivos, por este motivo estão sendo automatizados. A vantagem disso é obter precisão nos resultados, maior rapidez na produção, aumento de rendimento operacional, eficiência e baixo custo de mão de obra, assim como a aplicação industrial. É aplicada na colheita dos frutos, capina, poda, lavoura, irrigação, identificação de pragas, monitoramento da lavoura, aragem e pulverização.

Um dos grandes problemas para os agricultores é identificar áreas com incidência de pragas, aumentando seus gastos. Os **robôs agricultores** geram um mapa de produtividade, ajudando na precisão e reduzindo custos. Exemplo desse é o HortiBot, da Universidade de Aarhus, ele pode reconhecer mais de 25 tipos de ervas daninhas e eliminá-las utilizando os recursos de remoção.

Figura 33: HortiBot.



Fonte: Imagem tirada de (MIT, 2007).

Robôs feito por engenheiros da Universidade de Illinois e pela empresa EarthSense, chamado **TerraSentia**, têm função de examinar a plantação calculando o número de plantas, a largura do caule e a altura. O robô registra esses dados usando sensores como câmeras de vídeo, tecnologia de detecção, alcance de luz e navegação por GPS.

Figura 34: TerraSentia.



Fonte: Imagem tirada de (EarthSense, 2016).

Existem **robôs responsáveis pela irrigação** de lugares distintos. Certos robôs são programados para irrigar determinada área em determinado horário, que já são memorizados (figura 35(a)), há outros que são estufas modernas, no qual tudo pode ser controlado, a temperatura, umidade do ar, colocação de fertilizantes, irrigação e a radiação solar (figura 35(b)).

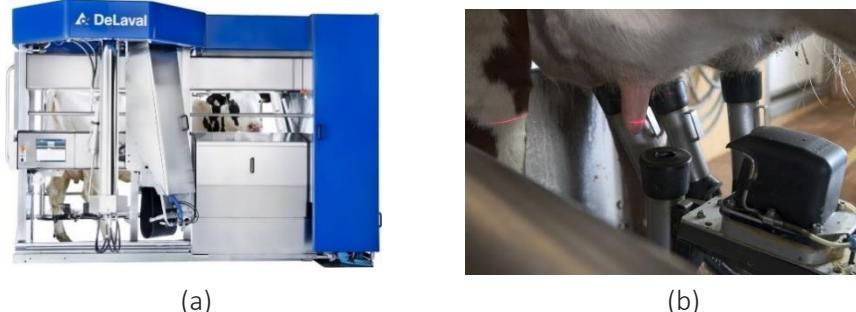
Figura 35: Robôs irrigadores: (a) robô irrigador e (b) estufa moderna.



Fonte: (a) Imagem tirada de (Robôs não Industriais UBI, 2013) e (b) Imagem tirada de (Robôs não Industriais UBI, 2013).

Já para pecuária está sendo desenvolvido protótipos de **robôs para vacinação**, evitando riscos de acidente para o homem. Além disso, existem também **robôs ordenhadores** que tiram leite das vacas, por meio de sensores e sugadores de leite a vácuo, toda essa informação é armazenada em um computador (quantidade extraída e histórico da vaca).

Figura 36: Robô Pecuária: (a) Robô Ordenhador e (b) sugadores de leite a vácuo em funcionamento.



Fonte: (a) Imagem tirada de (Leite & Corte, 2015) e (b) Imagem tirada de (MilkPoint, 2015).

CONCEITOS BÁSICOS

Introdução ao Arduíno

O que é Arduíno

O Arduíno é uma versátil plataforma de prototipagem eletrônica, de hardware e software aberto, de baixo custo e de fácil uso, mesmo para pessoas que possuem pouco conhecimento de eletrônica. (OLIVEIRA et al., 2018, p.13).

A placa Arduíno é semelhante a um computador, sendo composta por um microcontrolador, memória RAM, memória secundária, clock e comunicação USB. Além disso possui um software que consiste na linguagem C para Arduíno em um ambiente de desenvolvimento derivados do Processing.

Essa plataforma possui software livre, ou seja, um programa de computador que pode ser executado, copiado, modificado e redistribuído pelos usuários gratuitamente, com acesso ao código-fonte. A plataforma também trabalha com o conceito de hardware livre, onde é oferecido da mesma maneira que o software livre.

A família Arduino é composta por diferentes tipos de placas, dentre elas podem ser citadas: Arduino UNO, Arduino Mini, Arduino Nano, Arduino Mega, Arduino Duemilanove, Arduino Leonardo, Arduino Lilypad, Freeduino, Illuminato, Metaboard, Seeeduino, eJackino, Brasuino, Marminino etc. Durante os experimentos utilizaremos Arduíno UNO.

Figura 37: Arduino UNO.

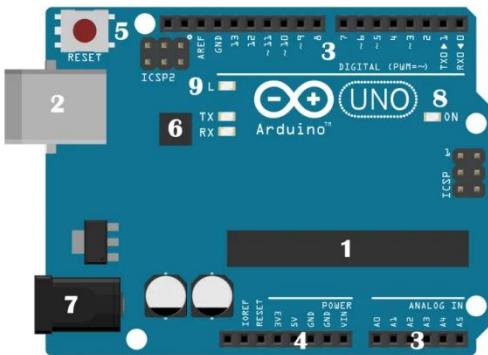


Fonte: Imagem tirada de (Arduino Portugal, 2017).

Hardware do Arduíno

Os principais elementos que compõem o hardware do Arduíno podem ser visualizados através da Figura 38.

Figura 38: Os principais elementos que compõem o hardware do Arduíno estão representados pelos números de 1 a 9



Fonte: Imagem tirada de (Arduino Portugal, 2017).

1: Microcontrolador ATMEL ATMEGA328, é a principal parte do Arduino, considerado seu cérebro, é um dispositivo programável que roda o código que enviamos à placa, este possui 32kB de memória flash e 2kB de memória RAM.

2: Alimentação é feita pela conexão USB.

3: Pinos de Entrada e Saída fazem com que o Arduino interaja com o meio externo, as portas são divididas em analógicas, digitais e PWM (explicados a seguir).

4: Conectores de alimentação IOREF (oferece uma tensão para que os Shields, placas que podem ser conectadas no Arduino, possam selecionar o tipo de interface apropriada, são alimentadas com 3,3V ou 5V), reset (é utilizado para reiniciar a placa Arduino), 3,3V (tem tensão de 3,3V para alimentação de Shields e módulos externos), 5V (tem tensão de 5V para alimentação de Shields e circuitos externos), GND (pino de referência, terra) e VIN (alimenta a placa através de Shields ou bateria externa).

5: Botão de Reset é o botão que reinicia o dispositivo.

6: Comunicação do computador com o USB é realizada pelo microcontrolador ATMEL ATMEGA16U2, é responsável pelo upload do código binário gerado após a compilação do programa feito pelo usuário. Nessa placa estão conectados dois leds (TX e RX), regidos pelo software do microcontrolador, que indicam o envio e recepção de dados do microcontrolador para o computador.

7: Alimentação do Arduino feita por meio externo.

8: Led de Alimentação mostra se a placa está transmitindo energia.

9: Led Interno Led ligado ao pino 13.

Portas Analógicas e Digitais

As **portas digitais** utilizam apenas dois valores lógicos, 0 para desligado ou 1 para ligado. No Arduino, esses estados do sistema binário são representados pelas tensões 0V (desligado) e 5V (ligado). A placa possui 14 portas digitais, sendo que algumas têm funções especiais.

As portas 3, 5, 6, 9, 10 e 11 podem ser utilizadas como saídas PWM, tratado mais a frente, simbolizados com um “~” na frente.

Os pinos 0 e 1 podem ser utilizados para comunicação serial, por isso é recomendável não se utilizar esses pinos. Estão ligados ao microcontrolador responsável pela comunicação do USB com o PC (RX e TX).

As portas 2 e 3 podem ser configuradas para gerar uma interrupção externa, através da função attachInterrupt().

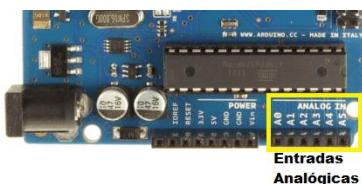
Figura 39: Pinos Digitais.



Fonte: Imagem tirada de (Embarcados, 2013).

As portas **Analógicas**, diferentemente das digitais, podem assumir valores entre 0V a 5V, que serão convertidos para valores entre 0 até 1023. A placa possui 6 pinos analógicos. Um dos principais componentes eletrônicos analógicos que conhecemos é o potenciômetro, nele podemos alterar sua resistência interna de forma manual.

Figura 40: Pinos Analógicos.



Fonte: Imagem tirada de (Embarcados, 2013).

As portas **PWM** são digitais, mas simulam portas analógicas, uma vez que podem gerar uma tensão média com valores entre 0V e 5V. As portas PWM são usadas para aplicações como: controle de velocidade de motores, variação da luminosidade de LEDs, geração de sinais analógicos e de áudio. Elas são representadas na placa Arduino com o sinal de um til (~).

Figura 41: Pinos PWM



Fonte: Imagem tirada de (Embarcados, 2014).

Instalação e Explicação do Software

Instalação

O download da última versão do software do Arduino, pode ser realizado por meio deste link: <http://www.arduino.cc/en/Main/Software>. Após isso descompacte o arquivo guardando esta pasta no drive C: do seu computador. Nesta pasta existe um arquivo chamado arduino.exe que é o ponto de entrada do programa do Arduino, a IDE (Integrated Development Environment).

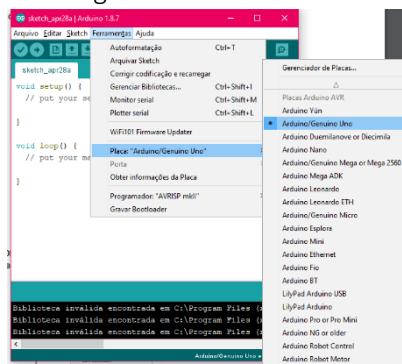
Conecte a sua placa Arduino ao computador com o cabo USB, a placa está conectada quando o Led estiver ligado, então será iniciado a instalação do driver, o processo vai falhar por

conta de ainda não se ter os drivers, então clique em concluir. Para obter êxito na instalação siga estes passos:

1. Clique no Menu Principal e abra o Painel de Controle;
2. Dentro do Painel de Controle, navegue até Sistema e Segurança;
3. Clique em Sistema, selecione Hardware e clique em Gerenciador de Dispositivos;
4. Procure por Portas (COM & LPT), onde deve-se ver uma opção Arduino UNO (COMxx);
5. Clique com o botão da direita em Arduino UNO (COMxx) e escolha a opção Atualizar Driver;
6. Escolha a opção Instalar de uma lista ou local específico (Avançado), e clique em avançar;
7. Escolha o driver arduino.inf que está na pasta Drivers do software do Arduino que foi baixado;
8. O Windows vai finalizar a instalação do driver.

Após a instalação do driver abra a IDE e vá até a aba Ferramentas>Placa e selecione a versão de sua placa, no caso Arduino UNO.

Figura 42: IDE Arduino.



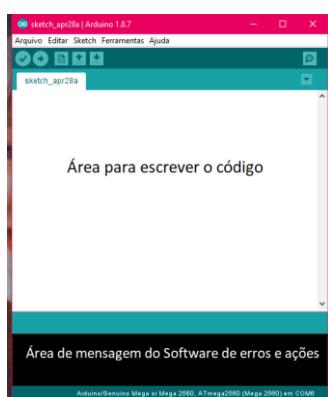
Fonte: Elaborado pela autora.

Selecione também a porta serial que conectará o Arduino: Ferramentas > Porta. Deve ser selecionado a mesma porta que se utilizou para configurar o sistema, quando foi feita a instalação de driver.

Explicação IDE

A IDE é muito simples e intuitiva, consistindo em um editor de código, um compilador e um depurador.

Figura 43: Espaço de Trabalho.



Fonte: Elaborado pela autora.

Os Sketches são softwares escritos utilizando Arduino, são salvos com a extensão de arquivo. ino.



Carregar

Compila seu código e carrega para a placa Arduino.



Verificar

Verifica se seu código apresenta erros.



Novo

Cria um novo Sketch



Abrir

Mostra um menu de todos os sketches que existem.



Salvar

Salva seu Sketch.

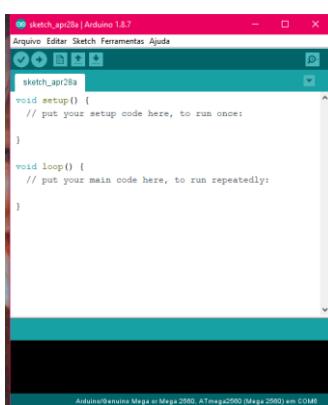


Monitor Serial

Abre o monitor serial.

No Arduino, um programa é chamado de sketch, e apresenta duas funções principais: setup() e loop().

Figura 44: Ambiente de Desenvolvimento do Arduino.



Fonte: Elaborado pela autora.

Nessas duas funções, a palavra void a acompanha, indicando que elas não apresentam um valor de retorno, e são utilizadas para a execução de uma série de comandos.

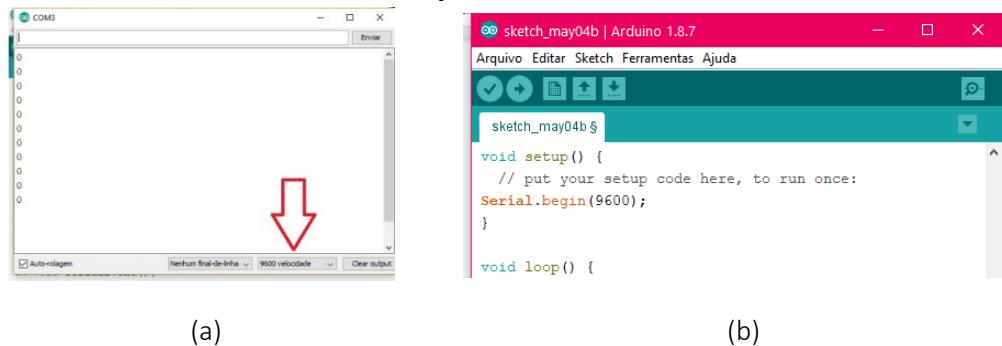
A função setup() deve ser formada pelo código que irá executar apenas uma vez. Ocorre quando o sketch inicia, sendo composta normalmente por definições iniciais do programa (pinos, LEDs, botão, entre outros).

A função loop() deve ser formada pelo código que será executado continuamente, até que seja carregado outro programa na memória “flash” do Arduino, sendo que um substitui o outro, sem poder salvar dois sketches de uma vez. Quando desligamos o Arduino o programa ainda estará armazenado, ou seja, quando ligado novamente executará normalmente o sketch salvo.

Monitor Serial

O monitor serial é uma janela que abre no IDE do Arduino, ela permite a exibição de dados enviados da placa Arduino para o computador. Pode acontecer em diversas velocidades pré-definidas, feitas de duas formas: pelo programa (por meio da função `Serial.begin(velocidade)`) ou pelo próprio monitor serial.

Figura 45: Definição de velocidade do monitor serial: (a) definição pelo monitor serial e (b) definição no sketch.



(a)

(b)

Fonte: (a) Elaborado pela autora e (b) Elaborado pela autora.

O Monitor Serial possui várias funções para comunicação, sendo algumas delas listadas na tabela 1.

Tabela 1: Funções do monitor serial.

Função	Significado
available()	Obtém o número de bytes disponíveis para leitura na porta serial.
begin()	Configura a taxa de dados em bauds para transmissão serial de dados.
end()	Desabilita a comunicação serial, permitindo que os pinos TX e RX (0 e 1) sejam usados para entrada e saída geral de sinais.
print()	Imprime dados na porta serial em formato legível (texto ASCII).
println()	Idem a print(), porém adicionando um caractere de retorno de carro (ASCII 13) e um de nova linha (ASCII 10) no final da string.
read()	Lê dados a partir da porta serial.
readBytes()	Lê caracteres a partir da porta serial em um buffer até que uma determinada quantidade de caracteres tenha sido lida.
write()	Escreve dados binários na porta serial.

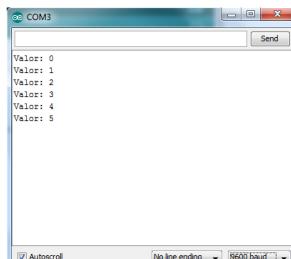
Tabela retirada do site: <http://www.bosontreinamentos.com.br/electronica/arduino/enviando-e-recebendo-dados-com-serial-monitor-no-arduino/>.

Para a melhor compreensão sobre o assunto faça este teste em na IDE observando o funcionamento do monitor serial.

```
/* Testando a saída serial do Arduíno  
Vamos escrever números na porta serial*/  
  
void setup(){  
    Serial.begin(9600); //enviar e receber dados em 9600 de velocidade  
    Serial.println("Gerando valores sequenciais");  
}  
  
int numero = 0;  
  
void loop() {  
    Serial.print("Valor: ");  
    Serial.println(numero);  
    delay(500);  
    numero++;  
}
```

Depois de digitar o código, compile-o e envie-o, então abra o monitor serial para verificar os dados sendo transmitidos do Arduino para seu PC.

Figura 46: Dados transmitidos pelo monitor serial.



Fonte: Imagem tirada de (Bósón treinamentos, 2015).

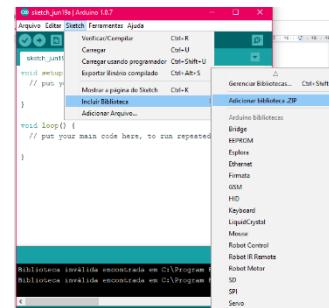
Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Biblioteca

Bibliotecas são coleções de códigos desenvolvidas com o objetivo de permitir o uso de algum hardware específico (MURTA, 2016). Existem centenas de bibliotecas na Internet para download, no caso do exemplo, será incluída a biblioteca do sensor de temperatura NTC, mas para qualquer biblioteca o processo será o mesmo. Siga o passo a passo:

1. Instale a biblioteca encontrada via Internet;
2. Clique em Sketch, Incluir Biblioteca e depois Adicionar biblioteca .ZIP;

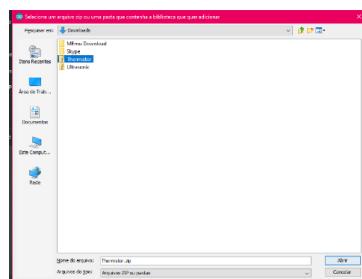
Imagen 47: Incluindo biblioteca.



Fonte: Elaborado pela autora.

3. Encontre a Biblioteca, a selecione e clique em Abrir.

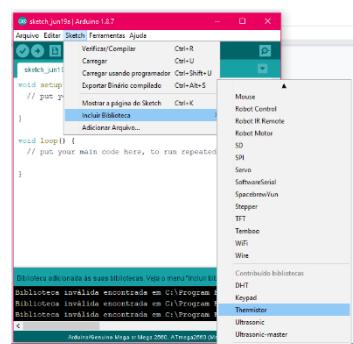
Imagen 48: Incluindo biblioteca.



Fonte: Elaborado pela autora.

4. A biblioteca já estará junto com as outras

Imagen 49: Biblioteca já incluída.



Fonte: Elaborado pela autora.

Conceitos básicos de eletrônica

Resistores e Lei de Ohm

Resistor é um dispositivo elétrico que tem um valor conhecido de resistência medido em ohm (Ω) e é utilizado para limitar a quantidade de corrente que pode fluir através dele. (WARREN; ADAMS; MOLLE, 2019, p.18).

Figura 50: Resistores.



Fonte: Imagem tirada de (Guia estudo, 2019).

A resistência é medida em ohm e nos informa a capacidade com que um condutor leva electricidade. O fluxo de corrente e a resistência são inversamente proporcionais, logo um condutor com menos resistência carrega mais electricidade.

Para calcular a potência dissipada pelo resistor, precisa-se saber a tensão do circuito e o valor da resistência em ohm. Assim, precisamos utilizar a lei de Ohm para definir a corrente que passará pelo resistor.

$$\begin{aligned} \text{Lei de Ohm: } V &= I \cdot R \\ I &= V/R \end{aligned}$$

V - Tensão ou potencial elétrico;

R - Resistência elétrica;

I - Corrente elétrica.

Exemplo: se tivermos uma resistência de 1.000 ohms (1K ohms) e uma fonte de alimentação de 12V.

$$I = 12V / 1.000 \text{ ohm}$$

$$I = 0,012 \text{ A ou } 12 \text{ mA}$$

A resistência pode ser calculada através das características do material resistivo:

$$R = \rho l / A$$

ρ - Resistividade do material;

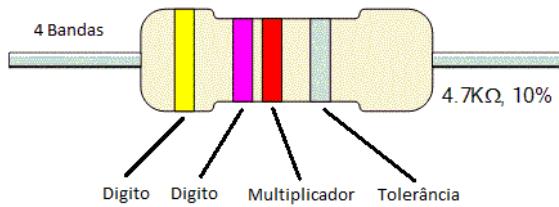
l - Comprimento do material;

A - Área do material.

Código de Cores

Um resistor é formado por várias cores (Figura 46), para que possamos determinar o valor de sua resistência. O código de cores é analisado através de faixas, tendo cada faixa sua função.

Figura 51: Ordem das faixas de cores.



Fonte: Imagem tirada de (FVW Learning, 2018)

Figura 52: Tabela de valores das cores.

Cor	Dígitos Significativos (1 e 2)	Multiplicador (3)	Tolerância (4)
Preto	0	10^0	
Marrom	1	10^1	$\pm 1\%$
Vermelho	2	10^2	$\pm 2\%$
Laranja	3	10^3	
Amarelo	4	10^4	
Verde	5	10^5	
Azul	6	10^6	
Violeta	7	10^7	
Cinza	8	10^8	
Branco	9	10^9	
Ouro	-	10^{-1}	$\pm 5\%$
Prata	-	10^{-2}	$\pm 10\%$
Sem Cor	-	-	$\pm 20\%$

Fonte: Imagem tirada de (Bóson Treinamentos, 2015).

1ª Faixa: É o primeiro número do valor da resistência.

2ª Faixa: É o segundo número do valor da resistência.

3ª Faixa: É o multiplicador, determina qual a ordem da resistência (mil, milhão etc.).

4ª Faixa: Determina a tolerância do resistor.

OBS: O resistor de 3 faixas não possui a faixa da tolerância, assim considera a ele uma tolerância de 20%.

$$1^{\text{a}} \text{ e } 2^{\text{a}} \text{ faixa juntas} \times 3^{\text{a}} \text{ faixa} = \text{resistência} (\Omega) \text{ e tolerância} (\%)$$

Exemplo: determinar o valor da resistência do resistor em ohms e a sua tolerância.

Figura 53: Resistor.



Fonte: Imagem tirada de (Athos Electronics, 2019);

1ª Faixa: Azul - 6;

2ª Faixa: Verde - 5;

3ª Faixa: Laranja - $\times 1000\Omega$ (10^3);

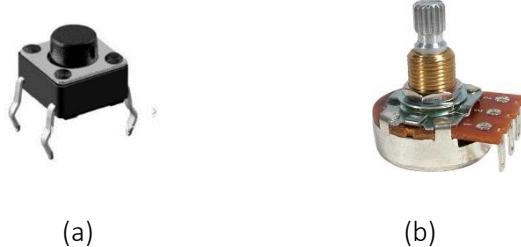
4ª Faixa: Dourado - $\pm 5\%$.

$$65 \times 1000\Omega (\pm 5\%) = 65000\Omega \pm 5\%$$

Entrada e Saída

As entradas, também chamadas inputs, são sensores que transformam os sinais do mundo em sinais de corrente ou voltagem. Existem entradas digitais e analógicas.

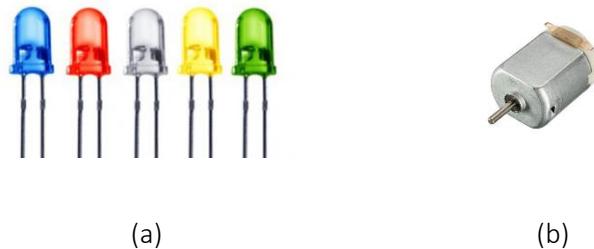
Figura 54: Exemplo de entradas: (a) Push-button (entrada digital) e (b) Potenciômetro (entrada analógica).



Fonte: (a) Imagem tirada de (Squids, 2018) e (b) Imagem tirada de (Bóson Treinamentos, 2018).

As saídas, também chamadas outputs, são dispositivos que convertem os sinais de corrente ou voltagem em sinais fisicamente úteis, faz o papel inverso da entrada. Existem saídas digitais e analógicas.

Figura 55: Exemplo de saídas: (a) Led (saída digital) e (b) Motor dc (saída analógica).



Fonte: (a) Imagem tirada de (Baú da Eletrônica, 2017) e (b) Flip Flop, 2020).

Multímetro

O multímetro é um aparelho que serve para medir grandezas elétricas. Existem modelos analógicos e digitais. Na robótica serve para medir tensão AC e DC, resistência, corrente DC, testar a continuidade, diodos, hFE de transistores, testar LEDs, entre outros.

Figura 56: Multímetro e suas funções na Robótica.



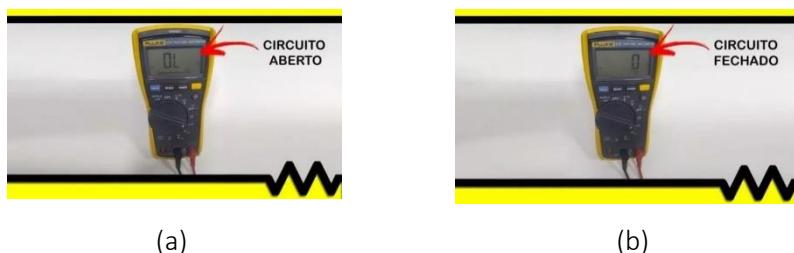
Fonte: Imagem tirada de (AECX, 2016).

Continuidade

O teste de continuidade nos permite verificar se existe ou não continuidade nas ligações dos condutores. Caso não exista, nada acontece, mas se existir, o multímetro emite um sinal sonoro, utilizamos para verificar se um jumper está rompido, para esta verificação siga os seguintes passos:

1. Ajuste a escala do multímetro para a escala de continuidade, representado por um símbolo semelhante ao de Wi-Fi;
2. Observe o display, quando não existe continuidade a leitura indica infinito ou OL, nenhuma corrente é capaz de fluir por este circuito;
3. Observe o display, quando existe continuidade a leitura indica o valor zero, o multímetro então emite um sinal sonoro de alerta, significando que o circuito ou cabo possui continuidade, não está rompido.

Figura 57: Teste de Continuidade: (a) não existe continuidade (OL) e (b) existe continuidade (0).



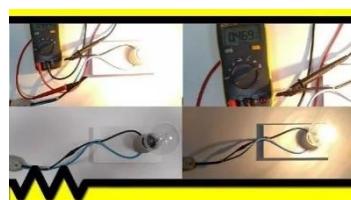
Fonte: (a) Imagem tirada de (Mundo da elétrica, 2014) e (a) Imagem tirada de (Mundo da elétrica, 2014).

Corrente Elétrica

A corrente elétrica é uma grandeza da eletricidade, sendo que sua unidade de medida é amperes (A). Ela pode ser contínua ou alternada, ou seja, pode variar o seu valor e sentido em função do tempo. Para medir a corrente com o multímetro, siga os passos abaixo:

1. Rotacione o seletor até chegar na escala de corrente elétrica, que é representado pelo símbolo (A);
2. Conecte os cabos, o preto no borne preto (COM), e o vermelho no borne vermelho, representado com um (A) de amperes e informando a máxima corrente elétrica suportada pelo multímetro;
3. As pontas do multímetro devem ser ligadas em série a um circuito desligado;
4. Faça um corte no condutor, para obter duas pontas, chamadas A e B;
5. Coloque a ponta de prova vermelha na ponta A do condutor e a ponta de prova preta na ponta B do condutor e isole;
6. Energize o circuito, obtendo a corrente elétrica que passa neste condutor;
7. Lembre-se sempre de emendar o corte feito e isolá-lo.

Figura 58: Circuito Energizado



Fonte: Imagem tirada de (Mundo da elétrica, 2014).

Tensão de Corrente Alternada

Também chamada de voltagem, a tensão de corrente alternada é a energia que tem nas tomadas elétricas, como por exemplo, é medida em V (volts). Para medi-la siga os seguintes passos:

1. Conecte os cabos, o preto no borne preto (COM), e o vermelho no borne vermelho, representado com um V;
2. Gire a chave seletora para função 750V se sua energia é acima de 200 volts, senão gire para a opção 200V do multímetro;
3. Dê contato nas partes que deseja medir, um pino na fase e outro no neutro do dispositivo que se deseja saber a voltagem, assim mostrará a tensão no display do multímetro.

Tensão de Corrente Contínua

Essa tensão serve para medir por exemplo pilhas e baterias, para medir siga os seguintes passos:

1. Gire a chave seletora do multímetro para função tensão de corrente contínua (DCV);
2. Coloque os pinos nas extremidades do objeto que for testado, colocando o cabo vermelho na extremidade positiva (+) e o cabo preto na negativa (-);
3. Quanto mais baixo apresentar os números no display, mais fraco estará o dispositivo testado.

Resistência

A resistência mede a dificuldade no percurso dos elétrons ao passar em um objeto, é medida em ohms, utilizamos para medir a resistência de um resistor. Para medi-la siga os seguintes passos:

1. Escolha um objeto que você deseja medir a resistência;
2. Conecte os cabos, o preto no borne preto (COM), e o vermelho no borne vermelho, representado com um Ω de ohms;
3. Ligue o multímetro e selecione a melhor amplitude (número junto com Ω) para testes, caso você não saiba qual a melhor amplitude, comece colocando nas menores amplitudes e toque no outro terminal, caso apareça OL no visor, aumente a amplitude;
4. Toque os terminais do multímetro nas extremidades do componente a ser testado, o número apresentado será sua resistência.

Figura 59: Amplitudes do multímetro.



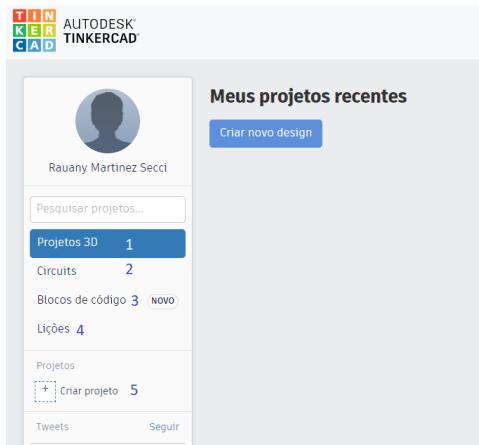
Fonte: Imagem tirada de (wikiHow, 2014).

TINKERCAD

Tinkercad é um programa online gratuito de modelagem 3D, suas principais características são simplicidade e facilidade de uso, funcionando em um navegador da web ou pode ser adquirido gratuitamente na Loja do Windows 10. É acessado a partir do link: <https://www.tinkercad.com/>.

Ao entrar no site crie sua conta, logo após será apresentada uma interface com os seguintes elementos:

Figura 60: Estrutura Tinkercad.



Fonte: Elaborado pela autora.

Projetos 3D (1)

Sua função é armazenar os projetos que são 3D e por meio do botão “criar novo design” abre uma página para formação de novos projetos.

Figura 61: Aba projetos 3D.



Fonte: Elaborado pela autora.

Os projetos 3D foram criados para fornecer uma maneira simples e gratuita de criar designs 3D, pois ferramentas desse tipo são caras e complicadas de aprender.

Figura 62: Nova janela de design projetos 3D



Fonte: Elaborado pela autora.

1. Renomear o projeto;
2. Por meio do “viewcube” podemos observar a saída de diferentes formas: left (esquerda), right (direita), top (em cima), front (frente), bottom (embaixo) e back (atrás);
3. Servem para observar a saída em diferentes perspectivas. De cima para baixo: retorna à posição e orientação padrão, visualização de todos os objetos, aumentar, diminuir e alternar entre perspectiva e projeção ortográfica;
4. Sua função é mostrar todos os elementos ocultos (selecionar + Ctrl + H);
5. Chamado de operação união, sua função é agrupar objetos, permitindo combiná-las em uma única forma (Ctrl + G);
6. Usado para desagrupar as formas que foram agrupadas anteriormente (Ctrl + Shift + G);
7. Sua função é alinhar objetos;
8. Sua função é espelhar objetos;
9. Apresenta o que for selecionado no número 12, classificados em Tinkercad (formas básicas, texto e números, caracteres, conectores, espaço de hangout OMSI e fazendo em casa), shape generators (em destaque e todos), Circuits (montagens e componentes), printables (smithsonian, dinossauro e esqueleto) e you (favoritos e coleção de peças);
10. Alterar a grade do plano de trabalho;
11. Plano de trabalho e a régua, que apresenta as dimensões do objeto no plano de trabalho;
12. Como citado no tópico 9, sua função é mostrar as categorias dos objetos existentes;

Figura 63: Janela de categorias dos objetos existentes.



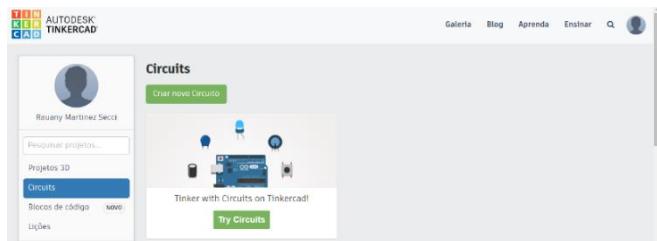
Fonte: Elaborado pela autora.

13. Da esquerda para direita, copiar, colar, duplicar e excluir;
14. Desfazer e refazer os últimos passos;
15. Plano de trabalho, onde aparece tudo o que está sendo executado.

Circuits (2)

Sua função é semelhante a aba anterior, mas ao invés de armazenar projetos 3D, armazenará circuitos; utilizaremos desse para montar os circuitos com Arduino. Por meio do botão “criar novo circuito”, abre uma página para formação de novos experimentos.

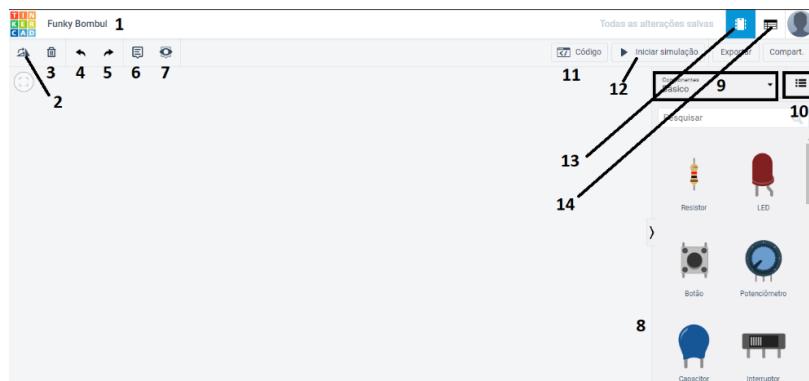
Figura 64: Aba Circuits.



Fonte: Elaborado pela autora.

Circuits é o módulo de eletrônica do sistema, que mantém a ideia da mão na massa, selecionando e interligando os componentes, formando um circuito eletrônico.

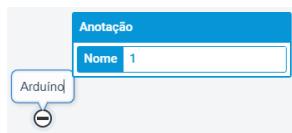
Figura 65: Nova janela de design Circuits.



Fonte: Elaborado pela autora.

1. Renomear o projeto;
2. Gira o elemento selecionado em 25 graus, chamado botão rotacional;
3. Exclui o objeto selecionado;
4. Desfaz o último passo;
5. Refaz o último passo;
6. Realiza uma anotação;

Figura 66: Exemplo de Anotação.



Fonte: Elaborado pela autora.

7. Exibe ou oculta uma anotação;

8. Onde apresenta todos os componentes eletrônicos, basta selecioná-lo e arrastar para a tela onde será montado o circuito. São classificados em básicos, todos, Arduino e montagem de circuito. Existe a possibilidade de pesquisar o componente desejado, digitando em pesquisar;
 9. Onde está classificado o tópico 8;

Figura 67: Janela de categorias dos componentes existentes.



Fonte: Elaborado pela autora.

10. Ao clicar neste botão, aparece uma breve explicação sobre cada item que está apresentado no tópico 8;

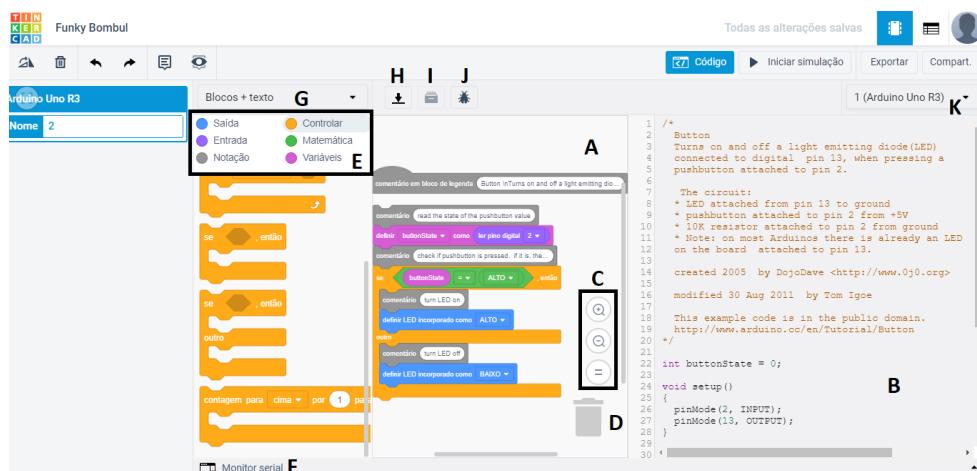
Figura 68: Exemplo de explicação sobre os componentes.



Fonte: Elaborado pela autora.

11. Esse botão é onde será digitado o código em C, para o funcionamento do circuito:

Figura 69: Janela de programar código.



Fonte: Elaborado pela autora

- A. Caso a programação seja feita por blocos, basta arrastá-lo para esse ambiente;
- B. Caso a programação seja feita em código, escreva o código neste ambiente;
- C. Aproxime (+), distancie (-) ou apresente os blocos de forma ideal (=);
- D. Remove os blocos indesejados;
- E. Sua função é montar o programa com blocos, cada classificação tem uma serventia;

Saída: definir as saídas, como a luz de um Led, imprimir no monitor serial ou reproduzir um som;

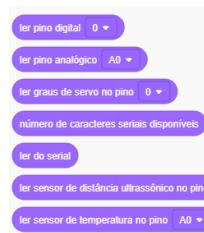
Figura 70: Tipos de saída.



Fonte: Elaborado pela autora.

Entrada: leitura de pinos, distância ou número do serial;

Figura 71: Tipos de entrada.



Fonte: Elaborado pela autora.

Notação: adicionar um comentário;

Figura 72: Tipos de notação.



Fonte: Elaborado pela autora.

Controlar: repetir uma ação, delimitando quantas vezes ou quanto tempo deve ser executado;

Figura 73: Tipos de controle.



Fonte: Elaborado pela autora.

Matemática: permite gerar números aleatórios, somar ou subtrair de uma variável, comparar dois valores (maior e menor, E, OU e NÃO);

Figura 74: Tipos de blocos matemáticos.



Fonte: Elaborado pela autora.

Variáveis: criar variável.

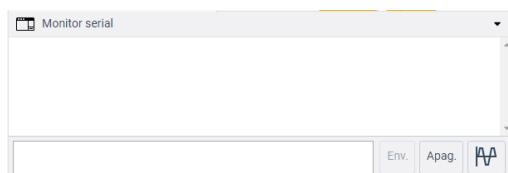
Figura 75: Tipos de variáveis.



Fonte: Elaborado pela autora.

F. Monitor Serial permite a exibição de dados enviados da placa Arduino para o computador;

Figura 76: Interface do monitor serial.



Fonte: Elaborado pela autora.

- G. Opção onde será decidido como será a programação do circuito (código + bloco, bloco ou código);
- H. Baixa o código feito no Tinkercad, no formato Arduíno;
- I. Chamado botão libraries, utilizamos quando o projeto é mais complexo, assim economizamos trabalho de codificar além do necessário.
- J. Depurador, serve para encontrar erros no programa;

- K. Seleciona o tipo de Arduíno que está sendo utilizado.
12. Para ver o resultado do programa, clique no botão “start simulation” uma emulação do que foi feito será iniciada;
 13. Mostra o status do projeto, ou seja, se foi salvo automaticamente;
 14. Forma uma espécie de tabela com todos os elementos utilizados no projeto.

Figura 77: Exemplo tabela com elementos do projeto.

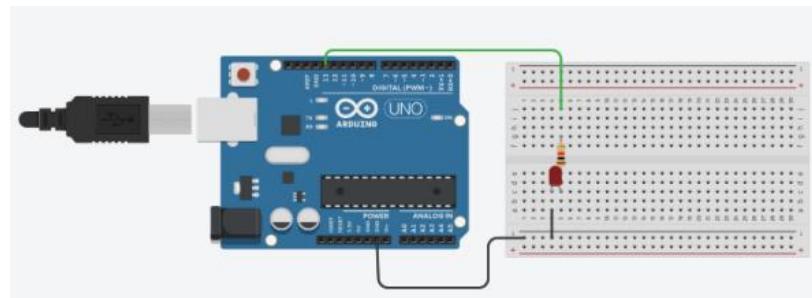
Nome	Quantidade	Componente
U2	1	Arduino Uno R3
R3	1	10 kΩ Resistor
S2	1	Botão

Fonte: Elaborado pela autora.

Agora para o treino, realize o seguinte experimento:

Crie o circuito abaixo no Tinkercad:

Figura 78: Experimento.



Fonte: Elaborado pela autora.

E copie o código fonte base:

```
int led = 13;
void setup() {
    pinMode(led, OUTPUT);
}
void loop() {
    digitalWrite(led, HIGH);
    delay(1000);
    digitalWrite(led, LOW);
    delay(1000);
}
```

Atividades:

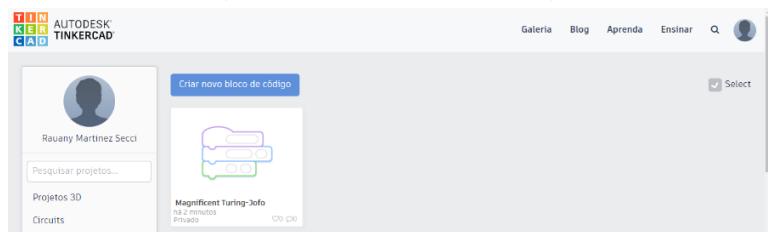
1. Altere o programa base para que o Led pisque a cada 2 segundos e fique 0,5 segundos desligados;
2. Acrescente um Led (Led 2) e um resistor na porta 12 do circuito.
 - a. Faça que o Led 2 pisque junto com o Led 1;
 - b. Alterne o acendimento dos Led 1 e 2 (enquanto o 1 estiver aceso o 2 deve ficar apagado).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Blocos de Código (3)

Uma ferramenta criada recentemente, têm a função de armazenar os blocos de código. Por meio do botão “criar novo bloco de código”, abre uma página para formação de novos experimentos.

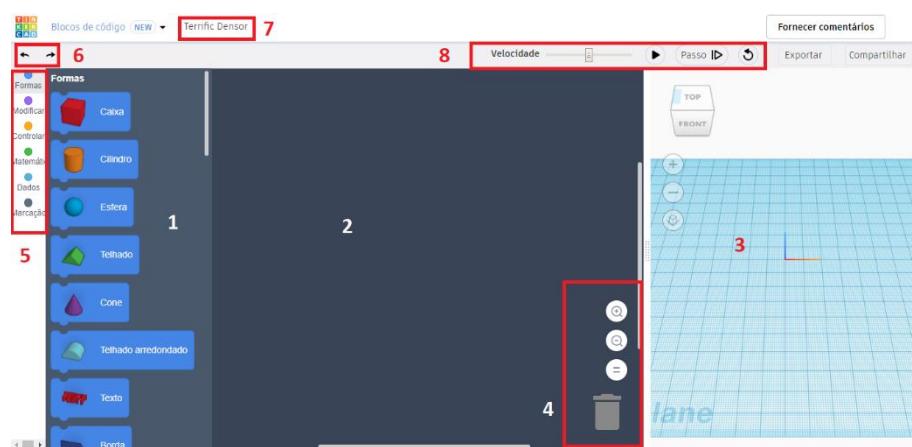
Figura 79: Aba blocos de código.



Fonte: Elaborado pela autora.

Bloco de códigos é o local mais simples para criar projetos 3D utilizando blocos de código arrastáveis.

Figura 80: Nova janela de design bloco de códigos.

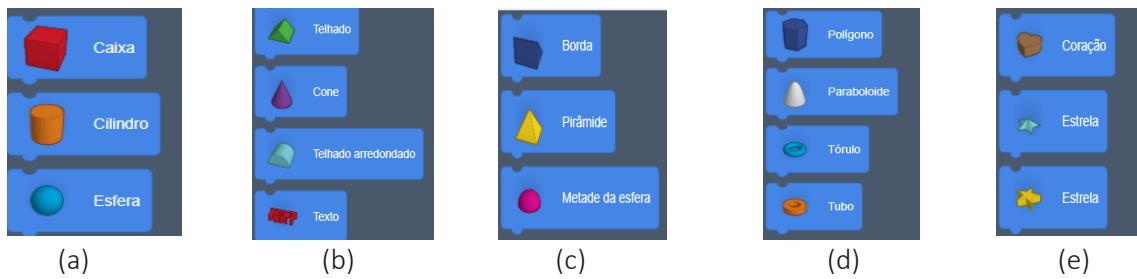


Fonte: Elaborado pela autora.

1. Os blocos de código servem para formar o programa, existem uma variedade de opções, sendo elas as seguintes:

- A. Formas: formado por todas as formas básicas do Tinkercad;

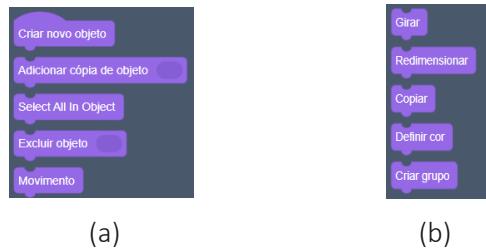
Figura 81: Formas: (a) caixa, cilindro e esfera, (b) telhado, cone, telhado arredondado e texto, (c) borda, pirâmide e metade da esfera, (d) polígono, paraboloide, tórulo e tudo e (e) coração e estrela.



Fonte: (a) Elaborado pela autora, (b) Elaborado pela autora, (c) Elaborado pela autora, (d) Elaborado pela autora e (e) Elaborado pela autora.

- B. Modificar: permitem alterar as formas existentes, como copiar, excluir, tamanho, cor, posição e rotação;

Figura 82: Modificar: (a) criar novo objeto, adicionar cópia de objeto, select all in object, excluir objeto e movimento e (b) girar, redimensionar, copiar, definir cor e criar grupo.



Fonte: (a) Elaborado pela autora e (b) Elaborado pela autora.

- C. Controlar: permitem definir loops para repetir ações, determinando quantas vezes ou quanto tempo a forma deve executar uma ação;

Figura 83: Controlar.



Fonte: Elaborado pela autora.

- D. Matemática: permitem definir variáveis, mudar valores, definir X e Y e gerar números aleatórios (random between);

Figura 84: Matemática.



Fonte: Elaborado pela autora.

- E. Dados: lista tudo o que já foi criado durante o programa;

Figura 85: Dados.



Fonte: Elaborado pela autora.

- F. Marcação: permitem adicionar comentários, mensagens e saída de texto.

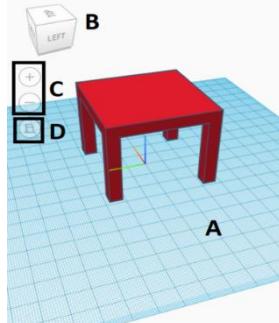
Figura 86: Marcação.



Fonte: Elaborado pela autora.

2. Este é o lugar onde será posto os blocos, formando uma combinação que seria seu “programa escrito”, por este motivo não precisa ter nenhum conhecimento de alguma linguagem de programação;
3. Por meio deste visualizador em 3D, quando você executa o código, a saída é visualizada;

Figura 87: Visualizador 3D.



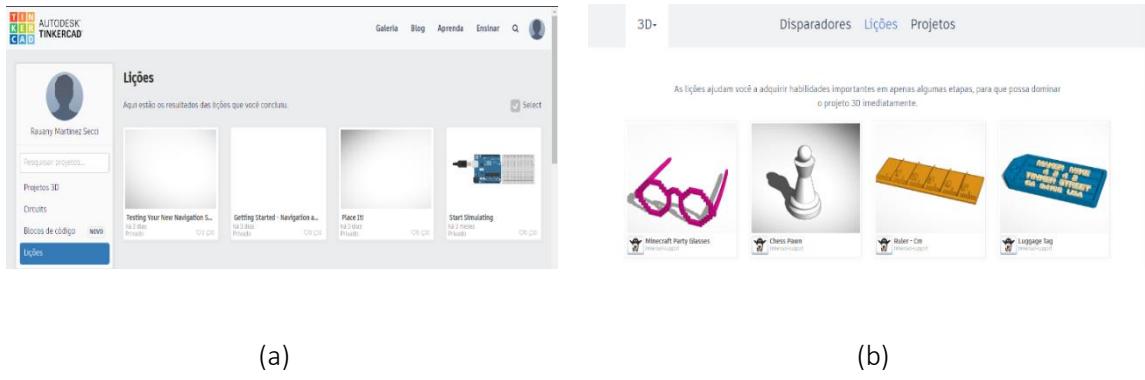
Fonte: Elaborado pela autora.

- A. Ambiente onde será apresentado a saída do seu programa, no caso do exemplo, uma mesa;
 - B. Por meio do “viewcube” podemos observar a saída de diferentes formas: left (esquerda), right (direita), top (em cima), front (frente), bottom (embaixo) e back (atrás);
 - C. É para observarmos a saída de mais perto (+) ou mais distante (-);
 - D. Mostra a saída em diferentes perspectivas.
4. Aproxime (+), distancie (-) ou apresente os blocos de forma ideal (=). A função da lixeira é remover os blocos indesejáveis;
 5. Estes botões têm função de encontrar rapidamente algum bloco de códigos, sendo classificados conforme a função do bloco (formas, modificar, controlar, matemática, dados e marcação);
 6. Desfazer e refazer os últimos passos;
 7. Renomear o projeto;
 8. Executar o programa. Da esquerda para direita: alterar a velocidade com que o programa está sendo executado, iniciar a execução do programa, avançar um passo e reiniciar.

Lições (4)

Nesta aba encontramos as lições que estão sendo realizadas ou as finalizadas. Elas podem ser adquiridas na aba aprenda, onde têm várias lições para aprender sobre o Tinkercad e suas ferramentas.

Figura 88: Lições: (a) aba lições e (b) lições na aba aprender.

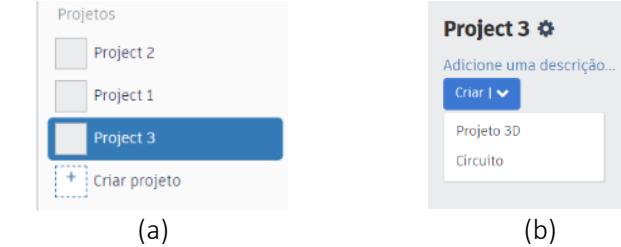


Fonte: (a) Elaborado pela autora e (b) Elaborado pela autora.

Criar Projeto (5)

Além dos botões que temos para criar projetos em Circuits e projetos 3D, temos essa aba que permite originar novos trabalhos. Clicando no '+ criar projeto', gera uma aba, onde será escolhido entre projeto 3D ou Circuits.

Figura 89: Criar projeto: (a) criando um novo projeto e (b) definindo qual é o tipo do projeto.



Fonte: (a) Elaborado pela autora e (b) Elaborado pela autora.

INTRODUÇÃO A LINGUAGEM C PARA ARDUINO

Estruturas

Estruturas de Controle

If / else

A estrutura condicional if/else permite diversos testes, onde se (if) a primeira expressão for verdadeira ela executará o bloco de comandos, mas senão (else) executará outro comando e assim por diante.

SINTAXE

```
if(condição) {  
    //bloco de comandos  
    comando1;  
    ...  
} else {  
    //bloco de comandos  
    comando2;  
    ...  
}
```

EXEMPLO

```
const int LED1 = 8;  
const int LED2 = 7;  
  
void setup() {  
    Serial.begin(9600);  
    pinMode(LED1, OUTPUT);  
    pinMode(LED2, OUTPUT)  
}  
  
void loop() {  
    if (Serial.available()) {  
        char valor = Serial.read();  
        if (valor == '1') {  
            digitalWrite(LED1, HIGH);  
            digitalWrite(LED2, LOW);  
        }  
        else {  
            digitalWrite(LED1, LOW);  
            digitalWrite(LED2, HIGH);  
        }  
    }  
}
```

While

O comando é executado infinitamente até que a condição seja falsa, para isso será necessário utilizar de um incremento ou decremento, caso contrário o loop será infinito.

SINTAXE

```
while (condição) {  
    //comandos  
    [incremento ou decremento da variável]  
}
```

EXEMPLO

```
int i=0;  
  
while (i<20){  
    digitalWrite (ledPin , HIGH);  
    delay (1000);  
    digitalWrite (ledPin , LOW);  
    delay (1000);  
    i++; //incremento  
}
```

Do-While

Semelhante ao While, mas sua diferença é que primeiro ele executa a ação e depois testa a condição.

SINTAXE

```
do{  
    // bloco de ações  
} while (teste de condição);
```

EXEMPLO

```
do{  
delay(100);  
x = readSensors(); // leitura de sensores, fará 100 leituras  
} while (x < 100);
```

Switch / case

Assim como if/else, é uma estrutura condicional. O switch compara a variável, que vem como parâmetro, com o valor de cada caso (case - determina diferentes códigos para cada caso, serão executados em diferentes condições), quando ele é encontrado a ação correspondente é feita.

SINTAXE

```
switch (var) {  
    case label:  
        // ação  
        break;  
    case label:
```

```

    // ação
    break;
default:
    // ação
}

EXEMPLO

switch (var) {
case 1:
    // executa os comandos quando var é igual a 1
    break;

case 2:
    // executa os comandos quando var é igual a 1
    break;

default:
    // Se nenhum dos anteriores, faz o caso padrão default
    // default não é obrigatório
    break;
}

```

For

Chamado de estrutura de laço, repete um número de vezes até que seja falso, é aplicado normalmente em operações com vetores. Seu parâmetro é composto por inicialização (variável ou valor que iniciará o laço), condição (se for verdadeira a ação é executada e a variável inicial é incrementada ou decrementada, se for falsa o laço termina) e incremento.

SINTAXE

```

for (inicialização; condição; incremento) {
    //ação;
}

```

EXEMPLO

```

int PWMpin = 10;

for (int i=0; i <= 100; i++){
    analogWrite(PWMpin, i);
    delay(10);
}

```

Break

Interrompe determinada condição de loop.

EXEMPLO

```

for (x = 0; x < 100; x ++){
    digitalWrite(PWMpin, x);
    sens = analogRead(sensorPin);

    if (sens > threshold){

```

```
    x = 0;  
    break;  
}
```

Continue

Ignora a interação de um loop a partir do ponto que ele se encontra, marcando a expressão condicional do loop e prosseguindo com o que vem depois.

EXEMPLO

```
for (x = 0; x < 100; x++) {  
  
    if (x > 35 && x < 110) {  
        continue;  
    }  
    digitalWrite(PWMPin, x);  
    delay(100);  
}
```

Return

Retorna um valor da função para a função de chamada.

SINTAXE

```
return;  
  
return valor;
```

Ambas estão certas, o valor pode ser alguma variável ou valor.

EXEMPLO

```
int checkSensor(){  
    if (analogRead(0) > 100) {  
        return 1;  
    }  
    else{  
        return 0;  
    }
```

Elementos de Sintaxe

Tabela 2: Elementos de Sintaxe.

Elemento	Função
;- ponto e vírgula	Encerra um comando.
{ } - chaves	São colocadas as instruções de uma função (setup, while, do-while, if/else, entre outras) dentro de chaves, delimitando onde começa e termina.

// - comentário	Adicionar comentário em uma linha, após // nada é considerado pelo programa.
/* */ - comentário em bloco	Adicionar comentário em várias linhas, entre /* e */ nada é considerado pelo programa.
#define	Atribuir um nome a um valor constante antes de o programa ser compilado.
#include	Incluir uma biblioteca externa.

Operadores

Aritméticos

Tabela 3: Operadores Aritméticos.

Símbolo	Significado
+	Soma.
-	Subtração.
*	Multiplicação.
/	Divisão.
%	Módulo (resto).
=	Operador de Atribuição.

Comparação

Tabela 4: Operadores de Comparação.

Símbolo	Significado
>	Maior que.
>=	Maior ou igual.
<	Menor que.
<=	Menor ou igual.
==	Igualdade.
!=	Diferente de.

Booleanos

Tabela 5: Operadores Booleanos

Símbolo	Significado
<code> </code>	OU lógico (or).
<code>&&</code>	E lógico (and).
<code>!</code>	NÃO lógico (not).

Composição

Tabela 6: Incremento e Decremento

Símbolo	Significado
<code>x++</code>	Soma 1 ao x e retorna o valor anterior de x.
<code>++x</code>	Soma 1 ao x e retorna o valor atual de x.
<code>x--</code>	Subtrai 1 ao x e retorna o valor anterior de x.
<code>--x</code>	Subtrai 1 ao x e retorna o valor atual de x.

Tabela 7: Operações Compostas.

Forma simplificada	Equivalência Usual
<code>x+=y</code>	$x=x+y$
<code>x-=y</code>	$x=x-y$
<code>x*=y</code>	$x=x*y$
<code>x/=y</code>	$x=x/y$

Valores

Constantes

Tabela 8: Constantes.

Construção	Descrição
<code>true</code>	1
<code>false</code>	0

HIGH	Pino digital com corrente (ligado).
LOW	Pino digital sem corrente (desligado).
INPUT	Entrada (só pode ser lido).
OUTPUT	Saída (só pode ser saída de dados).

Tipos de Dados

Tabela 9: Tipos de dados.

Tipo	Função
bool	Armazena o valor true ou false.
byte	Armazena valores de 0 a 255 de 8-bit sem sinal.
char	Armazena caracteres, quando for apenas um caracter utilizar aspas simples, caso contrário use aspas duplas.
double	Armazena números decimais, possuem 8-bytes.
float	Armazena números decimais, possuem 4-bytes.
int	Armazena números inteiros, possuem 2-bytes.
long	Variáveis de maior tamanho para armazenar números.
short	Tipo de dado 16-bit.
size_t	Representa o tamanho em bytes de determinado objeto.
string - char array	Armazenar um texto, ou por meio de aspas duplas, ou como vetor.
String object	- Constrói uma classe String.
unsigned char	Armazena valores de 0 a 255, mesmo tipo de dado que o byte.
unsigned int	O mesmo tipo de dado que o int, sua diferença é de só armazenar números positivos.
unsigned long	O mesmo tipo de dado que o long, sua diferença é de só armazenar números positivos.
vetor	Mesmo que array, armazena variáveis que são acessadas por um índice, pode ser definida com [], [variáveis que devem ser armazenadas] ou adicionando alguma palavra com aspas duplas.
void	Utilizada para declarar funções.

word	Armazena um número de no mínimo 16 bits.
------	--

Para mais detalhes acesse esse site: <https://www.arduino.cc/reference/pt/>.

Funções

Entrada e Saída Digital

Tabela 10: Funções de entrada e saída digital.

Construção	Descrição
pinMode(pino,modo)	Define o pino para funcionar como uma entrada ou saída (INPUT/OUTPUT).
digitalWrite(pino,valor)	Define um valor HIGH/LOW para o pino digital.
digitalRead(pino)	Lê o valor de um pino digital (HIGH/LOW).

Entrada e Saída Analógica

Tabela 11: Funções de entrada e saída analógica.

Construção	Descrição
analogReference(tipo)	Configura a tensão máxima da entrada analógica.
analogRead(pino)	Lê o valor de um pino analógico, valores entre 0 e 1023.
analogWrite(pino,valor)	Define um pino como PWM.

Temporização

Tabela 12: Funções de temporização.

Construção	Descrição
delay(ms)	Pausa o programa por um tempo definido em milissegundos.
millis()	Apresenta o tempo, em milissegundos, desde que a placa Arduino começou a executar o programa.
micros()	Apresenta o tempo, em microssegundos, desde que a placa Arduino começou a executar o programa.

Funções matemáticas

Tabela 13: Funções matemáticas.

Construção	Descrição
min(x, y)	Calcula o menor dos dois números dados como parâmetro.
max(x, y)	Calcula o maior dos dois números dados como parâmetro.
map(valor, deMenor, deMaior, paraMenor, paraMaior)	Mapeia um número de um intervalo para outro. EX: um valor de deMenor seria mapeado para paraMenor, um valor de deMaior para paraMaior.
abs(x)	Calcula o módulo do número dado como parâmetro.
pow(base, expoente)	Calcula o valor da base elevada ao expoente, dados como parâmetro.
constrain(x, a, b)	Faz com que o número (x) fique entre a (valor mínimo) e b (valor máximo).
sqrt(x)	Calcula a raiz quadrada do número dado como parâmetro.
sin(rad)	Calcula o seno do ângulo dado como parâmetro em radianos.
cos(rad)	Calcula o cosseno do ângulo dado como parâmetro em radianos.
tan(rad)	Calcula a tangente do ângulo dado como parâmetro em radianos.
random(min, max)	Gera números aleatórios, podendo ser determinado seu mínimo e máximo por meio do parâmetro.

Bits e bytes

Tabela 14: Funções de bits e bytes.

Construção	Descrição
bit(n)	Computa o valor do bit dado como parâmetro.
bitRead(x, n)	Lê o valor de um bit (n) em um número (x).
bitWrite(x, n, b)	Escreve (b é o valor escrito) um bit (n) de um valor numérico (x).
bitClear(x, n)	Limpa um bit (n) de uma variável numérica (x).
lowByte(x)	Apresenta o byte menos significativo da variável vinda do parâmetro.
highByte(x)	Apresenta o byte mais significativo de uma word vinda do parâmetro.

GUIA DA CAIXA DE ROBÓTICA

Fonte 9v Arduino Bivolt e Cabo USB

Por meio da fonte 9v Arduino bivolt (fonte externa) ou pelo cabo USB, alimentamos o Arduino.

Figura 90: Alimentação Arduino.



Fonte: Imagem tirada de (Embarcados, 2013).

Com a fonte 9v ao conectarmos na tomada, o código que anteriormente foi carregado a placa será executado.

Figura 91: Fonte 9v Arduino.



Fonte: Imagem tirada de (Submarino, 2020).

Ao conectar o cabo USB ao computador, além de alimentar a placa, pode-se compilar e transferir o código que será executado.

Figura 92: Cabo USB.

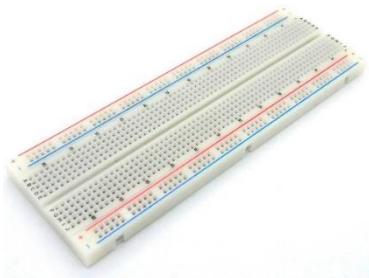


Fonte: Imagem tirada de (Autocore Robótica, 2020).

Protoboard

Sua função é conectar os componentes eletrônicos sem precisar soldá-los, possibilitando testes de diversos circuitos.

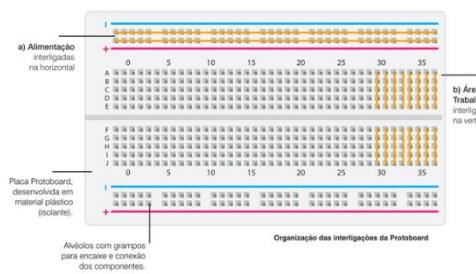
Figura 93: Protoboard.



Fonte: Imagem tirada de (Fronteira Tec, 2013).

A protoboard é composta por duas áreas distintas, uma para a alimentação elétrica (duas linhas superiores e inferiores sendo a vermelha para o positivo 5v e a azul para o negativo GND) onde se tem ligação direta com a placa Arduino, e outra para montagem dos circuitos (colunas).

Figura 94: Estrutura protoboard.



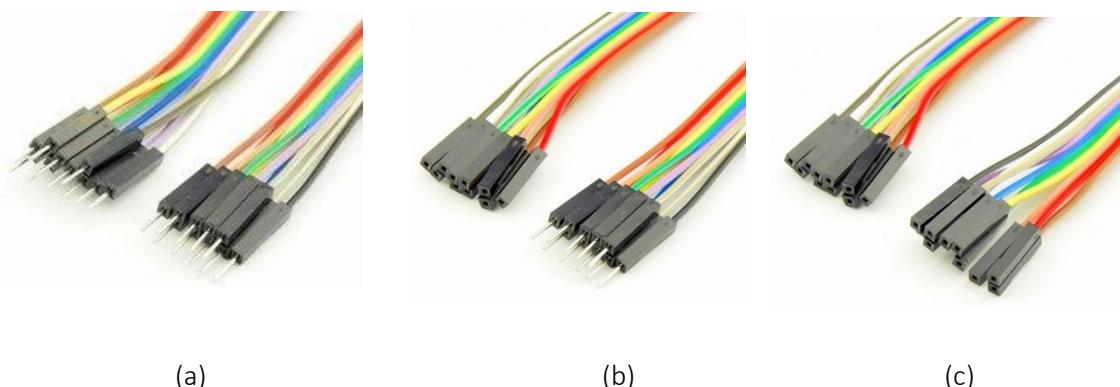
Fonte: Imagem tirada de (EAD IFFluminense, 2019).

Jumpers

Os jumpers são pequenos conectores revestidos de material isolante (NUSSEY, 2019). Sua função é interligar os componentes aos barramentos na protoboard, conectar a placa Arduino, entre outros elementos.

Eles podem ser classificados como: macho-macho, macho-fêmea e fêmea-fêmea.

Figura 95: (a) jumpers macho-macho, (b) jumpers macho-fêmea e (c) jumpers fêmea-fêmea.



Fonte: (a) Imagem tirada de (Recicomp, 2020), (b) Imagem tirada de (Recicomp, 2020) e (c) Imagem tirada de (Recicomp, 2020).

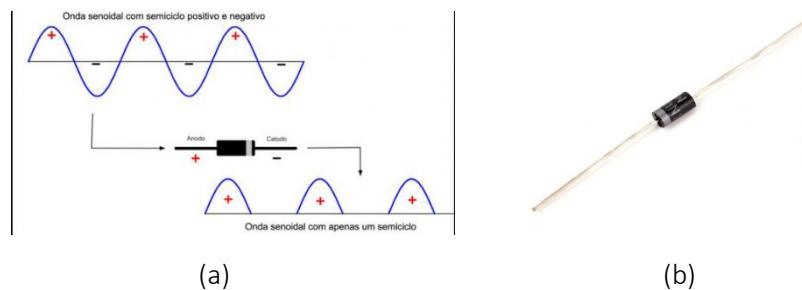
Resistor

Conforme escrito no Capítulo: Conceitos Básicos - Conceitos básicos da eletrônica - Resistores e Lei de Ohm/Código de cores.

Diodo

Sua principal função é fazer com que a corrente elétrica fique em apenas um sentido, fluindo do anodo (+) para o cátodo (-), assim a corrente se torna contínua, evitando de queimar algo do circuito caso a corrente inverta, se fosse ligada a uma fonte de corrente alternada.

Figura 96: (a) Funcionamento Díodo e (b) Díodo 1N4007.



Fonte: (a) Imagem tirada de (Mundo da elétrica, 2017) e (b) Imagem tirada de (Autocore, 2020).

Capacitor Cerâmico

É formado por duas placas metálicas condutoras separadas por um dielétrico de cerâmica, que se comporta como um condutor. A função de um capacitor é armazenar energia e liberá-la no processo de descarga. A capacidade de energia que ele armazena é chamada de capacitância.

O capacitor cerâmico é comum em circuitos de corrente contínua e alta frequência, guardando pouca quantidade de carga elétrica. Cada um deles possui uma espécie de codificação em números que representa seu valor.

Figura 97: Valor do capacitor cerâmico.



Fonte: Imagem tirada de (SomBox, 2014).

Ao observarmos o capacitor da imagem, podemos dizer que seu valor é 1 (primeiro dígito) 0 (segundo dígito) 0000 (4 zeros por conta do multiplicador ser 4), ou seja, 100000 pF.

Figura 98: (a) Estrutura do Capacitor Cerâmico e (b) Capacitor Cerâmico.



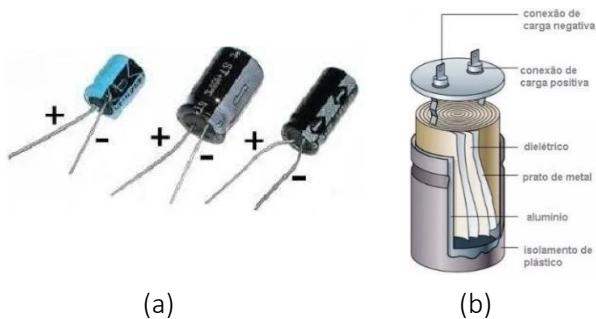
Fonte: (a) Imagem tirada de (Nova Eletrônica, 2017) e (b) Imagem tirada de (Nova Eletrônica, 2017).

Capacitor Eletrolítico

Assim como o capacitor cerâmico, o capacitor eletrolítico tem a função de armazenar energia e liberá-la no processo de descarga. Seu material isolante é menos espesso comparado aos demais capacitores, tendo camadas de material isolante e condutor enroladas umas nas outras.

O capacitor eletrolítico tem polaridade (positivo e negativo), indicada por uma marcação no capacitor, assim se for conectado de forma errada ele incha ou pode chegar a explodir.

Figura 99: (a) Capacitor eletrolítico e polaridade e (b) Estrutura capacitor eletrolítico.

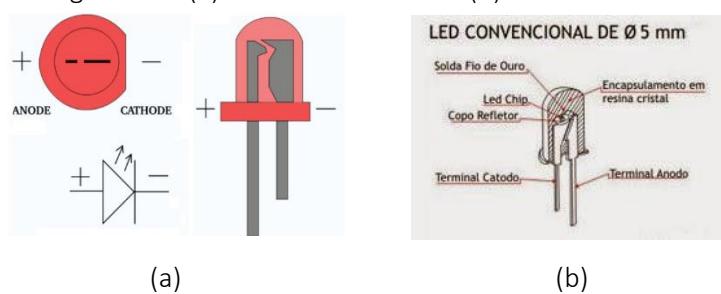


Fonte: (a) Imagem tirada de (Saber elétrica, 2015) e (b) Imagem tirada de (Saber elétrica, 2015).

LED

LED, no português diodo emissor de luz, tem a capacidade de transformar energia elétrica em luminosa. É um componente polarizado, ou seja, tem um lado positivo (anodo - maior lado) e um negativo (cátodo - menor lado). No Arduino, o positivo é conectado ao VCC e o negativo ao GND.

Figura 100: (a) Polaridade do LED e (b) Estrutura LED.



Fonte: (a) Imagem tirada de (Slideshare, 2012) e (b) Imagem tirada de (Dica da Arquiteta, 2015).

Praticando:

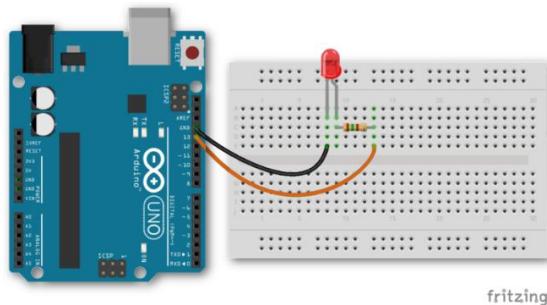
1- Acender LED.

Materiais necessários:

- Protoboard;
- 1 LED;
- 1 Resistor de 220 ohms;
- Jumpers;
- 1 Arduino Uno.

Círculo:

Figura 101: Círculo.



fritzing

Fonte: Imagem tirada de (FilipeFlop, 2018).

Código:

```
int led = 13;

void setup() {
    //Define a porta do led como saída
    pinMode(led, OUTPUT);
}

void loop() {
    //Acende o led
    digitalWrite(led, HIGH);
}
```

2- Piscar Led.

A diferença desse experimento com o anterior se encontra apenas no código, o LED para piscar deve acender e apagar.

Código:

```
int led = 13;

void setup() {
    //Define a porta do led como saída
    pinMode(led, OUTPUT);
}

void loop(){
    //Acende o led
    digitalWrite(led, HIGH);
```

```

    }
    delay(1000);
    //Apaga o led
    digitalWrite(led, LOW);
    delay(1000);
}

```

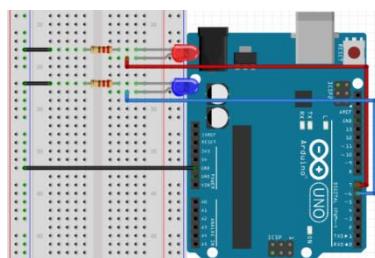
3- Intercalar a luz de dois LEDs.

Materiais necessários:

- 1 Protoboard;
- 2 LEDs;
- 2 Resistores de 220 ohms;
- Jumpers;
- 1 Arduino Uno.

Círculo:

Figura 102: Circuito.



Fonte: Imagem tirada de (Mundo projetado, 2017).

Código:

```

int led1 = 6;
int led2 = 7;

void setup() {
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
}

void loop() {
  digitalWrite(led1, HIGH);
  digitalWrite(led2, LOW);
  delay(1000);
  digitalWrite(led1, LOW);
  digitalWrite(led2, HIGH);
  delay(1000);
}

```

DESAFIO

Em cada um dos circuitos acima realize esses testes:

- Troque o LED por outro de cor diferente;

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;
 - Nos circuitos que o LED pisca mude o tempo que o LED pisca, alterando o delay (1000).

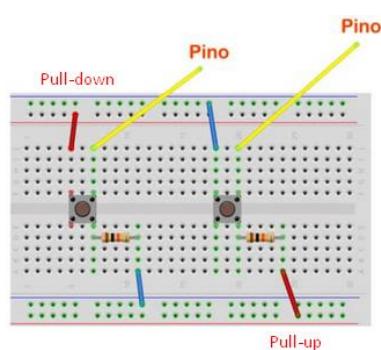
Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

Chave Táctil (Push-Button)

Um Push-button pode funcionar como uma entrada digital, onde só assume dois valores, quando o botão está sendo pressionado (conduzindo corrente elétrica, nível lógico 1) ou sem estar sendo pressionado (aberto, nível lógico 0).

Para garantir que os níveis lógicos sejam aproximados às tensões esperadas, deve-se utilizar os resistores ($10\text{K}\Omega$) pull-up e pull-down. Pull-up é conectado ao Vcc (+ = up) e Pull-down ao GND (- = down).

Figura 103: Pull-up e Pull-down



Fonte: Imagem tirada (Squids, 2018).

A função do resistor Pull-down é sempre receber o valor LOW (0) quando a chave estiver aberta.

Figura 104: Código Pull-down.

```


pullup


#define BOTAO 12

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

void loop()
{
    if (digitalRead(BOTAO) == HIGH)
        Serial.println("Botao pressionado!");
}

```

Fonte: Imagem tirada de (SHIGUEMORI; ARRAIS, 2019).

Observe que o botão é dado como pressionado quando o valor for HIGH.

A função do resistor Pull-up é sempre receber o valor HIGH (1) quando a chave estiver aberta. As portas digitais possuem pull-up, e para ativá-la basta definir como entrada e ligado (HIGH), pode ser ativado também por meio da constante INPUT_PULLUP.

```

void setup() {
    pinMode(10, INPUT_PULLUP);
}

```

Figura 105: Código Pull-up.

```


pullup


#define BOTAO 10

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

void loop()
{
    if (digitalRead(BOTAO) == LOW)
        Serial.println("Botao pressionado!");
}

```

Fonte: Imagem tirada de (SHIGUEMORI; ARRAIS, 2019).

Observe que o botão é dado como pressionado quando o valor for LOW.

Utilizando apenas os passos anteriores, o botão irá cometer erros causado pelo Efeito Bouncing, que faz com que o circuito entenda que o botão foi pressionado diversas vezes, ele dura em alguns milissegundos e pode ser resolvido de duas maneiras, pelo Hardware ou Software.

Por meio do Software a solução é adicionar um delay (em vermelho).

```

const int led = 2;
int Botao = 12;
int estadoBotao;

void setup(){
    pinMode(led, OUTPUT);
    pinMode(Botao, INPUT);
}

void loop(){
    estadoBotao = digitalRead(Botao); //lê o estado do botão - HIGH OU LOW
}

```

```

if (estadoBotao == HIGH){
    delay(100);
    estadoBotao = digitalRead(Botao);
    if (estadoBotao == HIGH){
        digitalWrite(led, HIGH);
    }
    else{
        digitalWrite(led, LOW); //deixa o led do pino 2 apagado
    }
}

```

Por meio do Hardware basta colocar um capacitor em seu circuito, que irá reduzir o sinal, gerando um tempo de atraso, assim as oscilações causadas pelo efeito Bouncing serão minimizados.

Praticando:

Não se esqueça de editar os códigos, caso necessário, para solucionar o Efeito Bouncing.

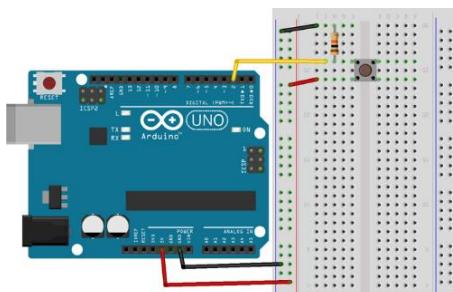
1- Lendo Botão com monitor Serial.

Materiais necessários:

- 1 Protoboard;
- 1 Push-button;
- 1 Resistor de 10K ohms e 1 Resistor de 220 ohms;
- Jumpers;
- 1 Arduino Uno.

Círculo:

Figura 106: Circuito.



Fonte: Imagem tirada de (Laboratório de garagem, 2014).

Código:

```

int botao = 2;
void setup() {
    pinMode(botao, INPUT);
    Serial.begin(9600);
}

void loop() {
    int buttonState = digitalRead(botao);
    if (buttonState == 1) {
        Serial.println("ligado");
    }
}

```

```

    } else {
        Serial.println("desligado");
    }
}

```

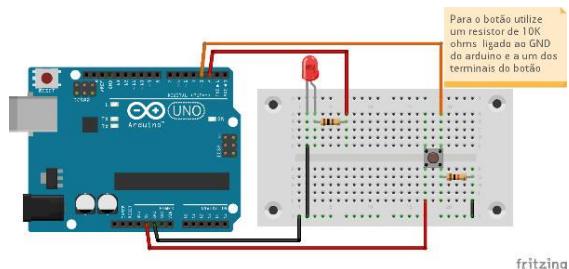
2- Ligando e desligando LED com botão.

Materiais necessários:

- 1 Protoboard;
- 1 LED;
- 1 Push-button;
- 1 Resistor de 10K ohms e 1 Resistor de 220 ohms;
- Jumpers;
- 1 Arduino Uno.

Círculo:

Figura 107: Círculo.



Fonte: Imagem tirada de (Pedro Hs, 2017).

Código:

```

int ledPin = 2;
int botao = 3;
int estbotao = 0;

void setup() {
    pinMode(ledPin, OUTPUT);
    pinMode(botao, INPUT);
}

void loop(){
    estbotao = digitalRead(botao);
    if (estbotao == HIGH) {
        digitalWrite(ledPin, HIGH);
    }
    else {
        digitalWrite(ledPin, LOW);
    }
}

```

DESAFIO

Em cada um dos circuitos acima realize esses testes:

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;

- Faça os testes de pull-up e pull-down;
 - Adicione mais um Push-button e modifique o código para ligar o LED com uma chave e desligar com a outra.

Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

Potenciômetro

É um componente que limita o fluxo de corrente elétrica sendo ajustada manualmente, podendo ter valores de 0 a 1023. Existem diversos formatos de potenciômetros, os que fazem parte da caixa de robótica são os de eixo giratório e o trimpot.

O potenciômetro de eixo giratório é o mais comum, sua resistência é ajustada girando o eixo, o encontramos em diversas aplicações, como em rádios e televisões.

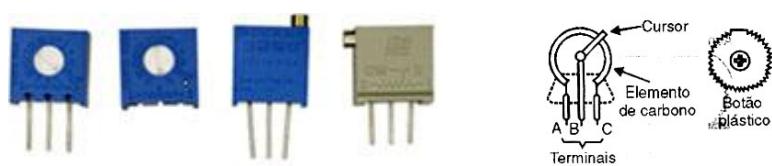
Figura 108: (a) Potenciômetro de eixo giratório e (b) Funcionamento potenciômetro de eixo giratório.

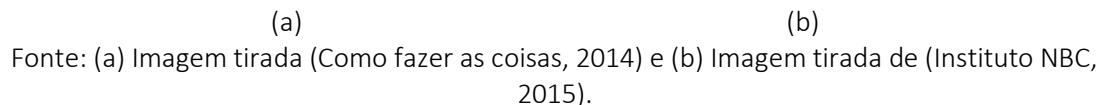


Fonte: (a) Imagem tirada (Como fazer as coisas, 2014) e (b) Imagem tirada de (YouTube, 2015).

O potenciômetro Trimpot é mais encontrado dentro de equipamentos eletrônicos, sua resistência é ajustada semelhante a apertar ou afrouxar um parafuso, necessitando de uma chave fenda ou Philips bem pequena.

Figura 109: (a) Potenciômetro Trimpot e (b) Funcionamento potenciômetro trimpot.





Praticando:

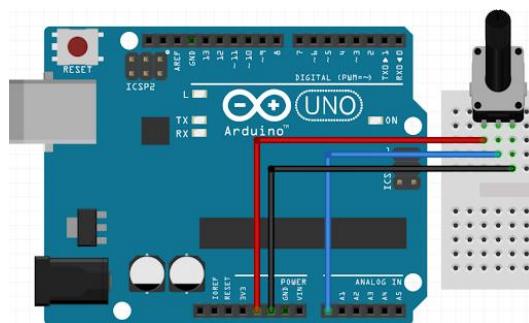
1- Lendo valores analógicos e apresentando-os no monitor serial.

Materiais necessários:

- 1 Protoboard;
- 1 Potenciômetro;
- Jumpers;
- 1 Arduino Uno.

Círculo:

Figura 110: Circuito.



Fonte: Imagem tirada de (Mundo projetado, 2017).

Código: Abra o monitor serial e gire o potenciômetro, assim será apresentado os valores.

```
const int potenciometro = A0; // pino de entrada do potenciômetro
int valor = 0;

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

void loop() {
    valor = analogRead(potenciometro);
    Serial.println(valor);
    delay(100);
}
```

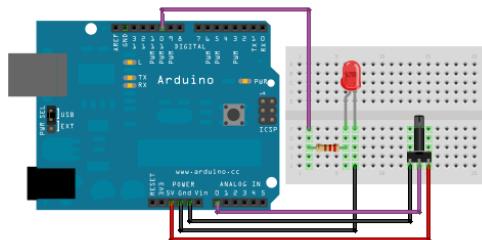
2- Controlando intensidade do LED com potenciômetro.

Materiais necessários:

- 1 Protoboard;
- 1 LED;
- 1 Resistor de 220 ohms;
- 1 Potenciômetro;
- Jumpers;
- 1 Arduino Uno.

Círcuito:

Figura 111: Círcuito.



Fonte: Imagem tirada de (Blog Do José Cintra, 2016).

Código:

```
#define potPin 0
#define ledPin 10
int valPot = 0;

void setup() {
    pinMode(ledPin,OUTPUT);
    pinMode(potPin,INPUT);
}

void loop() {
    valPot = analogRead(potPin); //Leitura do potenciômetro
    valPot = map(valPot,0,1023,0,255); //Transforma a escala
    analogWrite(ledPin,valPot ); // Aciona o LED proporcionalmente
}
```

DESAFIO

Em cada um dos circuitos acima realize esses testes:

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;
- Adicionar mais um LED no circuito 2.

Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

Sensor de Luz LDR

LDR é um componente eletrônico que muda a resistência conforme a luminosidade, de forma inversamente proporcional, ou seja, quanto maior a incidência de luz menor é a resistência. O LDR é ativado ou desativado por meio da quantidade de luz. Os seus terminais não têm polaridade.

Figura 112: Sensor de Luz LDR.



Fonte: Imagem tirada de (Mundo projetado, 2017).

Praticando:

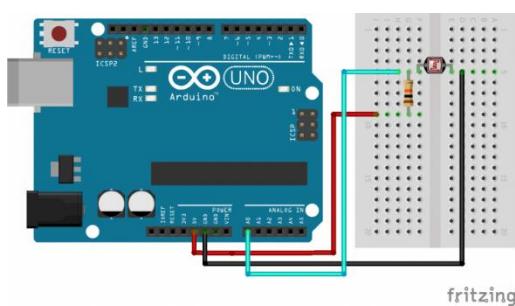
1- Leitura do sensor.

Materiais necessários:

- 1 Sensor LDR;
- 1 Arduino Uno;
- 1 Resistor 10K ohms;
- 1 Protoboard;
- Jumpers diversos.

Círcuito:

Figura 113: Circuito.



Fonte: Imagem tirada de (Athos Electronics, 2016).

Código:

```
int LDR = A0
int ValorLDR; //Vai armazenar a leitura do sensor
int IntensidadeLuz; //Transforma a leitura em uma escala de 0 a 100

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

```

void loop() {
    ValorLDR = analogRead(LDR); //Faz a leitura do sensor
    IntensidadeLuz = map(ValorLDR, 0, 1024, 1, 100); //Converte o valor para uma escala de 0 a 100
    Serial.print("Intensidade de Luz = "); //imprime o valor lido na tela
    Serial.println(IntensidadeLuz);
    delay(300);
}

```

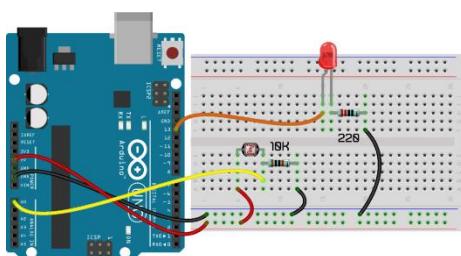
2- Ligar ou desligar o LED de acordo com a intensidade de luz presente no ambiente.

Materiais necessários:

- LED Vermelho 5 mm;
- 1 Resistor 220 ohms e 1 Resistor 10K ohms;
- 1 Sensor LDR;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Circuito:

Figura 114: Circuito.



Fonte: Imagem tirada de (Filipeflop, 2018).

Código:

```

int pinoLed = 13;
int pinoSensorLuz = A0;
int valorLuz = 0;
void setup() {
    pinMode(pinoLed,OUTPUT);
}
void loop() {
    valorLuz = analogRead(pinoSensorLuz);
    if(valorLuz<750) {
        digitalWrite(pinoLed,HIGH);
    }
    else {
        digitalWrite(pinoLed,LOW);
    }
    delay(10);
}

```

DESAFIO

Em cada um dos circuitos acima realize esses testes:

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;
- No circuito 2 mude a sensibilidade do sensor LDR modificando o valor na linha `if(valorLuz<750)`.
- No circuito 2 modifique a programação para realizar o inverso do objetivo proposto.

Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

Sensor de Temperatura NTC e PTC

Conhecidos como termistores, tem uma alta sensibilidade com mudança de temperatura. Variam a resistência dependendo da temperatura, no sensor NTC a resistência diminui com o aumento da temperatura, sendo inversamente proporcional, já o sensor PTC a resistência aumenta com o aumento da temperatura, ou seja, diretamente proporcional.

Figura 115: Sensor de Temperatura NTC.



Fonte: Imagem tirada de (Master Walker, 2019).

Praticando:

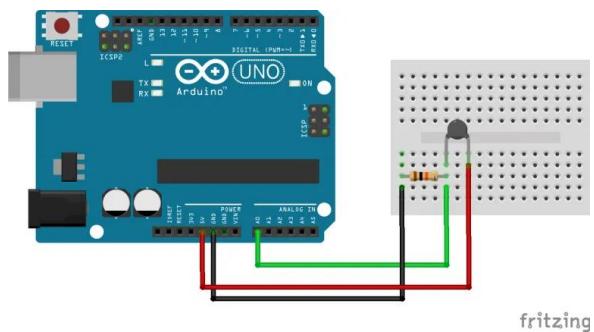
1- Leitura do sensor.

Materiais necessários:

- 1 Sensor NTC;
- 1 Arduino Uno;
- 1 Resistor 10K ohms;
- 1 Protoboard;
- Jumpers diversos.

Círcito:

Figura 116: Circuito.



Fonte: Imagem tirada de (Auto Core, 2017).

Código: Para o funcionamento deste código você terá que instalar essa biblioteca aqui.

```
#include <Thermistor.h>
Thermistor temp(0);

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

void loop() {
    int temperature = temp.getTemp();
    Serial.print("Temperatura: ");
    Serial.print(temperature);
    Serial.println("°C");
    delay(1000);
}
```

DESAFIO

No circuito acima realize esse teste:

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Buzzer

O buzzer é um componente que funciona a partir do efeito piezoelétrico, mas de modo contrário. O efeito piezoelétrico é caracterizado como o surgimento de uma tensão elétrica a partir de um esforço mecânico, no caso do buzzer acontece o inverso, o surgimento interno de uma tensão mecânica a partir de um campo elétrico aplicado, assim dentro do buzzer irá vibrar na mesma frequência, formando o som.

Ele é composto por dois terminais, maior positivo e menor negativo. Por meio de sua vibração, sua frequência é definida, “moldando” um determinado som.

Figura 117: Buzzer.



Fonte: Imagem tirada de (Seu Robô, 2013).

Praticando:

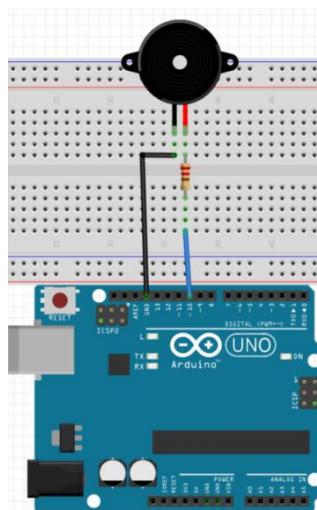
1- Emitindo som.

Materiais necessários:

- 1 Buzzer;
- 1 Arduino Uno;
- 1 Resistor;
- 1 Protoboard;
- Jumpers diversos.

Círcuito:

Figura 118: Circuito.



Fonte: Imagem tirada de (Mundo Projetado, 2017).

Código:

```
const int buzzer = 10;  
  
void setup() {
```

```

pinMode(buzzer,OUTPUT);
}

void loop() {
  tone(buzzer,1500); //Ligando o buzzer com uma frequência de 1500 hz.
  delay(500);
  noTone(buzzer); //Desligando o buzzer.
  delay(500);
}

```

2- Tocando música com buzzer.

A diferença desse experimento com o anterior se encontra apenas no código, então repita o mesmo circuito.

Código:

```

char buz = 10;
char* musica[] = {"La","Re","Fa","Sol","La","Re", "Fa", "Sol", "Mi", "Pausa", "Sol", "Do", "Fa", "Mi",
"Sol", "Do", "Fa", "Mi", "Re", "Fim"}; //Definindo as notas e pausas da música de Game of Thrones.
int duracao[] = {700, 500, 300, 250, 250, 300, 200, 200, 700, 200, 500, 500, 200, 200, 200, 500,
200, 200, 500};
char* starwars[] = {"La", "Pausa", "La", "Pausa", "La", "Pausa", "Fa", "Do", "La", "Pausa", "Fa", "Do",
"La", "Pausa", "Mi", "Pausa", "Mi", "Pausa", "Mi", "Pausa", "Fa", "Do", "Sol", "Pausa", "Fa", "Do",
"La", "Pausa", "La", "Pausa", "La", "Pausa", "La", "Pausa", "La", "Pausa", "Sol#", "Pausa", "Sol",
"Fa#", "Fa", "Fa#", "Fim"}; //Definindo as notas e pausas da música de Starwars - Marcha Imperial.
int dur[] = {400, 100, 400, 100, 400, 100, 300, 200, 300, 100, 300, 200, 300, 200, 400, 100, 400,
100, 400, 100, 300, 300, 200, 100, 300, 300, 200, 200, 400, 50, 400, 50, 400, 50, 400, 50, 300,
50, 300, 200, 200, 200};

void setup(){
  pinMode(buz, OUTPUT);
  tocar(musica,duracao);
  //tocar(starwars,dur);
}

void loop(){
  // Não utilizado
}

void tocar(char* mus[], int tempo[]){
  int tom = 0;
  for(int i = 0; mus[i]!="Fim";i++){
    if(mus[i] == "Do") tom = 262;
    if(mus[i] == "Re") tom = 294;
    if(mus[i] == "Mi") tom = 330;
    if(mus[i] == "Fa") tom = 349;
    if(mus[i] == "Sol") tom = 392;
    if(mus[i] == "La") tom = 440;
    if(mus[i] == "Si") tom = 494;
    if(mus[i] == "Do#") tom = 528;
    if(mus[i] == "Re#") tom = 622;
    if(mus[i] == "Fa#") tom = 370;
    if(mus[i] == "Sol#") tom = 415;
    if(mus[i] == "La#") tom = 466;
  }
}

```

```
if(mus[i] == "Pausa") tom = 0;  
tone(buz, tom, tempo[i]);  
delay(tempo[i]);  
}  
}
```

DESAFIO

Em cada um dos circuitos acima realize esses testes:

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;
- No circuito 1 mude a frequência do buzzer;
- Busque outros códigos com novas músicas e teste o buzzer.

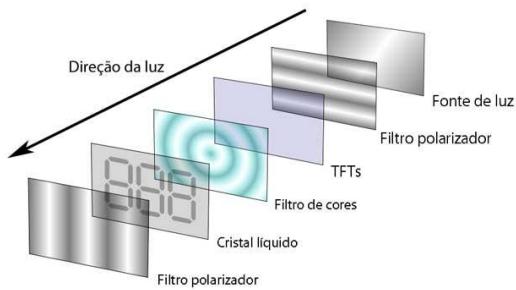
Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

Display LCD

LCD significa em inglês *Liquid Crystal Display*, traduzido mostrador de cristal líquido. Sua função é exibir a saída de dados, sem o uso de monitor serial ou outra forma de saída. Este cristal líquido se encontra comprimido entre duas lâminas transparentes polarizadoras (PIMENTA, 2014).

A luz passa pelo filtro polarizador, depois passa pela camada de TFT (Thin-Film Transistor), que ajuda no contraste e endereçamento de pixels, logo após pelo filtro de cores, onde acrescenta cores, chegando ao cristal líquido, onde visualizamos a saída.

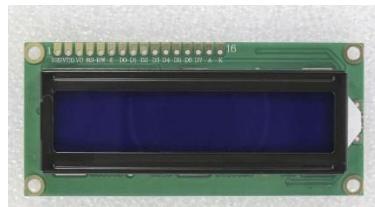
Figura 119: Camadas display LCD.



Fonte: Imagem tirada de (IBM, 2012).

O Display LCD encontrado na caixa de robótica é composto por 16 pinos com as seguintes funções.

Figura 120: Display LCD.



Fonte: Imagem tirada de (Eletrogate, 2018).

Figura 121: Funções pinos display LCD.

Pinos	Função	Conexão
1	Vss	GND
2	Vdd	Vcc 5V
3	VO	Pino central potenciômetro
4	RS	Pino 12 Arduino
5	RW	GND
6	E	Pino 11 Arduino
7	DO	Não conectado
8	D1	Não conectado
9	D2	Não conectado
10	D3	Não conectado
11	D4	Pino 5 Arduino
12	D5	Pino 4 Arduino
13	D6	Pino 3 Arduino
14	D7	Pino 2 Arduino
15	A	Vcc 5V
16	K	GND

Fonte: Imagem tirada de (SHIGUEMORI, 2019).

O ajuste do contraste e a regulagem da luz de fundo dos pinos 15 e 16 podem ser feitos por um trimpot ou potenciômetro. Para utilizá-lo deve ser incluída a biblioteca LiquidCrystal.h, que já estará instalada no software do Arduino, a partir dessa existem diversos comandos que podem ser utilizados para o controle do display LCD.

Tabela 16: Comandos para o LCD

Função	Sintaxe no programa	Descrição
LiquidCrystal()	LiquidCrystal(rs, enable, d4, d5, d6, d7)	Indica como os pinos do display estão interligados na placa Arduino.

begin()	lcd.begin(colunas, linhas)	Configura o modelo do display em nosso caso 16x2.
clear()	lcd.clear()	Limpa o display e posiciona o cursor no canto superior esquerdo do display.
home()	lcd.home()	Posiciona o cursor no canto superior esquerdo do display.
setCursor()	lcd.setCursor(coluna, linha)	Posiciona o cursor na linha e coluna indicada no comando.
write()	lcd.write(caractere)	Escreve um caractere no display.
print()	lcd.print(dado)	Escreve um dado qualquer no display (texto ou número).
cursor()	lcd.cursor()	Exibe o cursor na tela do LCD no formato de um underline.
noCursor()	lcd.noCursor()	Oculta o cursor.
blink()	lcd.blink()	Faz com que o cursor fique piscante na tela do Arduino.
noBlink()	noBlink()	Faz com que o display pare de piscar.
noDisplay()	lcd.noDisplay()	Desliga o display sem perder os dados exibidos.
display()	lcd.display()	liga o display e restaura o texto escrito após um comando "noDisplay()".
scrollDisplayLeft()	lcd.scrollDisplayLeft()	Move o conteúdo do display (texto+cursor) um espaço para a esquerda.
scrollDisplayRight()	lcd.scrollDisplayRight()	Move o conteúdo do display (texto+cursor) um espaço para a direita.
autoscroll()	lcd.autoscroll()	Ativa a rolagem automática do LCD. Isso faz com que cada caractere empurre os anteriores um espaço. Se a direção do texto atual é da esquerda para a direita (o padrão), o visor desloca para a esquerda; se a direção atual é da direita para a esquerda, o visor desloca para a direita.
noAutoscroll()	lcd.noAutoscroll()	Desliga o modo automático autoscroll.
leftToRight()	lcd.leftToRight()	Define a direção para o texto ser escrito no LCD da esquerda para a direita, o padrão. Isto significa que os caracteres subsequentes escritos para a exibição vão da esquerda para a

		direita, mas não afeta o texto enviado anteriormente para o display.
rightToLeft()	lcd.rightToLeft()	Define a direção para o texto ser escrito no LCD da direita para a esquerda. Isto significa que os caracteres subsequentes escritos para a exibição vão da direita para a esquerda, mas não afeta o texto enviado anteriormente para o display.
createChar()	lcd.createChar()	Cria um caractere personalizado (glyph) para uso no LCD. Até oito caracteres de 5x8 pixels são suportados (numerados de 0 a 7). O aparecimento de cada caractere personalizado é especificada por uma matriz de oito bytes, uma para cada linha. Os cinco bits menos significativos de cada byte determinam os pixels daquela linha. Para exibir um caractere personalizado na tela, use o comando lcd.write(byte(numero));

Tabela retirada do site: <http://blog.baudaelectronica.com.br/conhecendobibliotecaliquidcrystal/>.

Praticando:

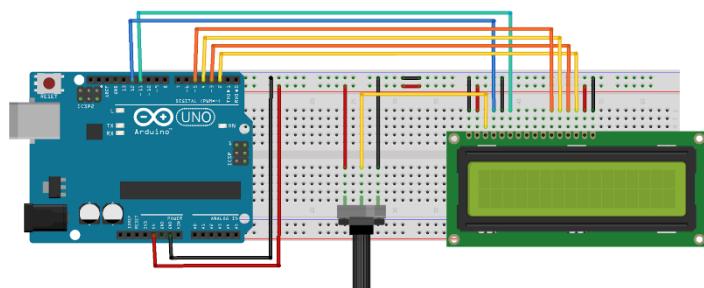
1- Controlando LCD.

Materiais necessários:

- 1 Display LCD;
- 1 Potenciômetro;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Círcuito:

Figura 122: Circuito.



Fonte: Imagem tirada de (Laboratório de Garagem, 2013).

Código:

```
#include <LiquidCrystal.h>
//Define os pinos que serão utilizados para ligação ao display
```

```

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
    //Define o número de colunas e linhas do LCD
    lcd.begin(16, 2);
}
void loop() {
    lcd.clear(); //Limpa a tela
    lcd.setCursor(3, 0); //Posiciona o cursor na coluna 3, linha 0
    lcd.print("HELLO"); //Envia o texto entre aspas para o LCD
    lcd.setCursor(3, 1);
    lcd.print("LCD 16x2");
    delay(5000);
}

```

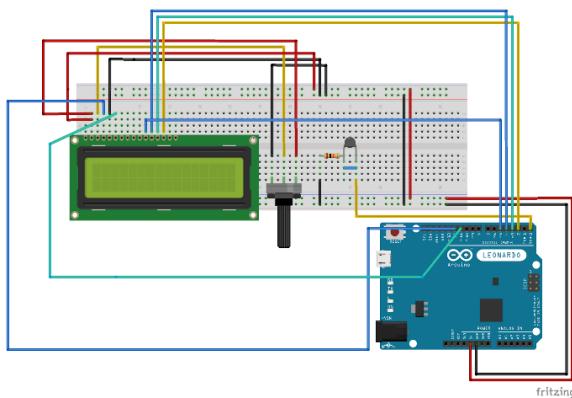
2- Termômetro e LCD.

Materiais necessários:

- 1 Sensor NTC;
- 1 Display LCD;
- 1 Potenciômetro;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos;
- 1 Resistor 10K ohms.

Círculo:

Figura 123: Círculo.



Fonte: Imagem tirada de (RoboCore, 2014).

Código: Para o funcionamento deste circuito, deve ser instalado a biblioteca do sensor de temperatura NTC, o link dessa biblioteca se encontra neste capítulo, no subcapítulo Sensor de temperatura ntc e ptc.

```

#include <Thermistor.h>
#include <LiquidCrystal.h>
Thermistor temp(1); // termistor conectado na porta A1 (cria o objeto)
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

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

```

```
lcd.begin(16, 2);
}

void loop() {
    float temperature = temp.getTemp(); //calcula a temperatura
    //Imprimindo no monitor serial
    Serial.print("Temperatura: ");
    Serial.print(temperature);
    Serial.println("°C");
    //imprimindo no LCD
    lcd.setCursor(0,0); // Define o cursor na posição de início
    lcd.print("Temp. = ");
    lcd.print(temperature);
    lcd.write(B11011111); // Imprime o símbolo de grau
    lcd.print("C");
    delay(1000);
}
```

DESAFIO

Em cada um dos circuitos acima realize esses testes:

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;
- No circuito 1 faça com que apareça seu nome no Display LCD;
- No circuito 2, adicione o Sensor de luminosidade LDR, para observar a temperatura e a luminosidade.

Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

Sensor de Distância Ultrassônico

O sensor Ultrassônico é capaz de detectar a distância de obstáculos que estão à sua frente, por este motivo é extremamente utilizado na frente de robôs ou equipamentos que precisam evitar colisão com leituras de distâncias entre 2 cm e 4 metros, com precisão de 3 mm, seu funcionamento é baseado no sonar dos morcegos.

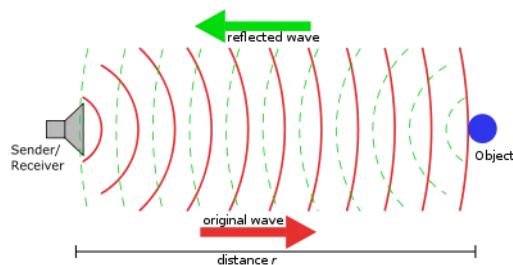
Figura 124: Sensor de distância ultrassônico.



Fonte: Imagem tirada de (Master Walker, 2018).

Por meio de uma onda sonora, imperceptível ao ouvido humano já que a onda tem uma frequência de 40khz (ultra som) e o ser humano consegue perceber sons na faixa de 20hz a 20khz, ele detecta um objeto que reflete essa onda, que retorna em forma de eco. Assim ele efetua o cálculo da distância, por meio do tempo que o sinal emitido levou para retornar ao sensor.

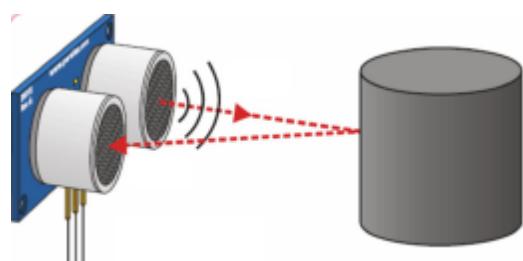
Figura 125: Cálculo da distância entre o sensor e o objeto.



Fonte: Imagem tirada de (AutoCore, 2017).

O cálculo para chegar até essa distância é distância percorrida entre o sensor e o objeto = (velocidade do som no ar (340 m/s)*tempo que o sinal emitido levou para retornar ao sensor)/2. A divisão por 2 ocorre pois o sensor mede o tempo de ida e volta, ou seja, duas vezes do tempo.

Figura 126: Sensor Ultrassônico emitindo ondas sonoras até entrar em contato com objeto.



Fonte: Imagem tirada de (SHIGEMORI, 2019).

Praticando:

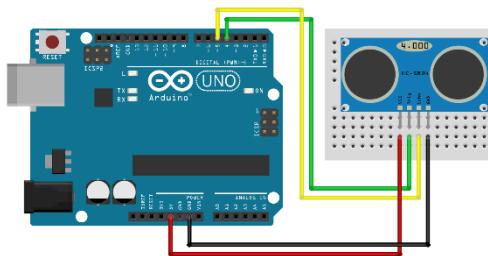
1- Calculando distância em cm e polegadas, apresentando no monitor serial.

Materiais necessários:

- 1 Sensor de distância ultrassônico;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Círcuito:

Figura 127: Circuito.



Fonte: Imagem tirada de (Filipeflop, 2011).

Código: Para o funcionamento deste código você terá que instalar essa biblioteca aqui.

```
#include <Ultrasonic.h>
#define pino_trigger 4
#define pino_echo 5
Ultrasonic ultrasonic(pino_trigger, pino_echo); //Inicializa o sensor nos pinos definidos acima

void setup(){
  Serial.begin(9600);
  Serial.println("Lendo dados do sensor...");
}

void loop(){
  //Le as informacoes do sensor, em cm e pol
  float cmMsec, inMsec;
  long microsec = ultrasonic.timing();
  cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
  inMsec = ultrasonic.convert(microsec, Ultrasonic::IN);
  Serial.print("Distancia em cm: ");
  Serial.print(cmMsec);
  Serial.print(" - Distancia em polegadas: ");
  Serial.println(inMsec);
  delay(1000);
}
```

2- Calculando distância em cm e polegadas, apresentando no display LCD.

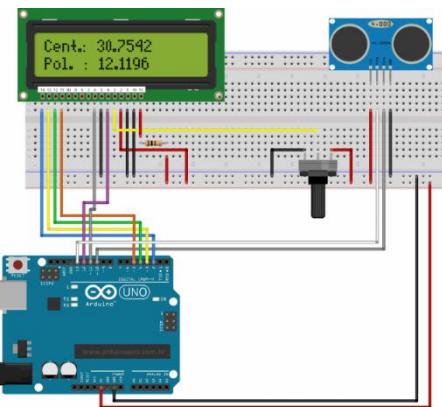
Materiais necessários:

- 1 Sensor de distância ultrassônico;
- 1 Display LCD;

- 1 Potenciômetro;
- 1 Resistor 10K ohms;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Círcuito:

Figura 128: Círcuito.



Fonte: Imagem tirada de (Arduino e CIA, 2013).

Código:

```
#include <Ultrasonic.h> //Carrega a biblioteca Ultrasonic
#include <LiquidCrystal.h> //Carrega a biblioteca LCD
#define PINO_TRIGGER 13
#define PINO_ECHO 10
Ultrasonic ultrasonic(PINO_TRIGGER, PINO_ECHO); //Inicializa o sensor ultrasonico
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //Define os pinos que serão ligados ao LCD

void setup() {
  Serial.begin(9600);
  lcd.begin(16,2);
  lcd.clear();
}

void loop() {
  float cmMsec, inMsec;
  long microsec = ultrasonic.timing(); //Le os dados do sensor, com o tempo de retorno do sinal
  cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM); //Calcula a distancia em centimetros
  inMsec = ultrasonic.convert(microsec, Ultrasonic::IN); //Calcula a distancia em polegadas
  lcd.setCursor(0,0);
  lcd.print("Cent.: ");
  lcd.print("    ");
  lcd.setCursor(7,0);
  lcd.print(cmMsec);
  Serial.print("Cent: ");
  Serial.print(cmMsec);
  lcd.setCursor(0,1);
  lcd.print("Pol. : ");
```

```

lcd.print("      ");
lcd.setCursor(7,1);
lcd.print(inMsec);
Serial.print(", Pol. : ");
Serial.println(inMsec);
delay(1000);
}

```

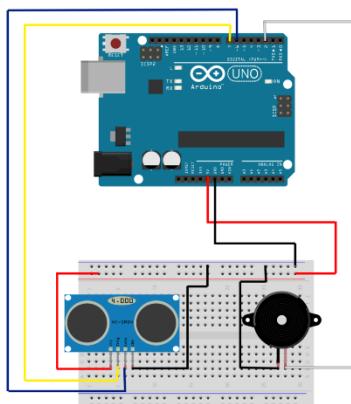
3- Alarme de distância com buzzer.

Materiais necessários:

- 1 Sensor de distância ultrassônico;
- 1 Buzzer;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Circuito:

Figura 129: Circuito.



Fonte: Imagem tirada de (Master Walker, 2016).

Código:

```

#include "Ultrasonic.h"
const int echoPin = 6;
const int trigPin = 7;
const int pinoBuzzer = 2;
Ultrasonic ultrasonic(trigPin,echoPin); //INICIALIZANDO OS PINOS
int distancia;

void setup(){
  pinMode(echoPin, INPUT); //DEFINE O PINO COMO ENTRADA (RECEBE)
  pinMode(trigPin, OUTPUT); //DEFINE O PINO COMO SAÍDA (ENVIA)
  pinMode(pinoBuzzer, OUTPUT); //DECLARA O PINO COMO SENDO SAÍDA
}

void loop(){
  hcsr04(); //chama a função hcsr04
  if(distancia <= 30){
    tone(pinoBuzzer,1500);//ACIONA O BUZZER
  }
}

```

```

}else{
    noTone(pinoBuzzer);//BUZZER PERMANECE DESLIGADO
}
}

void hcsr04(){
    digitalWrite(trigPin, LOW); //SETA O PINO 6 COM UM PULSO BAIXO "LOW"
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH); //SETA O PINO 6 COM PULSO ALTO "HIGH"
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW); //SETA O PINO 6 COM PULSO BAIXO "LOW" NOVAMENTE
    distancia = (ultrasonic.Ranging(CM));
    delay(500);
}

```

DESAFIO

Em cada um dos circuitos acima realize esses testes:

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;
- No circuito 1 e 2 adicione uma nova medida para ser lida (Ex: metro, milímetro etc.);
- No circuito 3 diminua a distância em que o buzzer será ativado, depois teste o inverso.

Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

Sensor Óptico Reflexivo

O sensor óptico reflexivo é composto por um coletor e um emissor, sendo o emissor um LED que emite luz infravermelha e o coletor um fototransistor que retém o feixe de luz emitido pelo LED, que assim que recebe essa luz é ativado. Normalmente é utilizado para detectar linhas de marcação. Sua distância máxima é de 2,5 cm.

Figura 130: (a) Sensor óptico reflexivo TCRT5000 e (b) Funcionamento do sensor.



(a)

(b)

Fonte: (a) Imagem tirada de (Eletrogate, 2017) e (b) Imagem tirada de (Arduino e CIA, 2013).

Praticando:

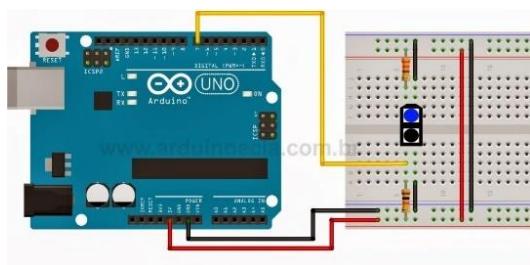
1- Conectando o sensor óptico reflexivo no Arduino.

Materiais necessários:

- 1 Sensor óptico reflexivo TCRT5000;
- 1 Resistor de 330 ohms e 1 Resistor de 10K ohms;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Círcuito:

Figura 131: Circuito.



Fonte: Imagem tirada de (Arduino e CIA, 2013).

Código:

```
int objeto = 0;

void setup() {
  pinMode(7, INPUT); //Pino ligado ao coletor do fototransistor
  Serial.begin(9600);
}

void loop() {
  objeto = digitalRead(7);
  if (objeto == 0){
    Serial.println("Objeto : Detectado");
  }
  else{
    Serial.println("Objeto : Ausente !");
  }
}
```

2- Interruptor Óptico (altera o estado do LED, cada vez que um objeto é detectado).

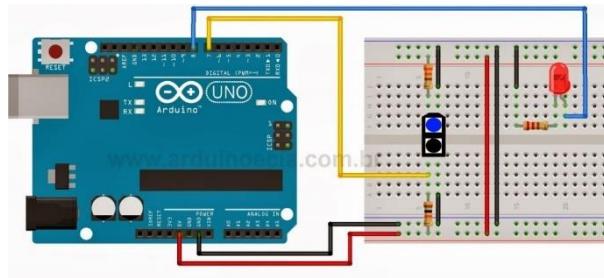
Materiais necessários:

- 1 Sensor óptico reflexivo TCRT5000;
- 1 LED;
- 2 Resistores de 330 ohms e 1 Resistor de 10K ohms;
- 1 Arduino Uno;
- 1 Protoboard;

- Jumpers diversos.

Círculo:

Figura 132: Círculo.



Fonte: Imagem tirada de (Arduino e CIA, 2013).

Código:

```
int sinalparaoled = 8; //Pino do led
int pinosensor = 7; //Ligado ao pino "coletor" do sensor óptico
int leitura;
int estandoled = 0;

void setup() {
pinMode(sinalparaoled, OUTPUT);
pinMode(pinosensor, INPUT);
}

void loop() {
leitura = digitalRead(pinosensor); //Le as informações do pino do sensor
if (leitura != 1) { //Verifica se o objeto foi detectado
while(digitalRead(pinosensor) != 1) {
delay(100);
}
//Inverte o estado do led (ligado / desligado)
estandoled = !estandoled;
digitalWrite(sinalparaoled, estandoled); //Liga ou desliga o led conforme "estandoled"
}
}
```

DESAFIO

Em cada um dos circuitos acima realize esses testes:

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;
- Teste até que distância o sensor está reconhecendo o objeto sem margem de erros;
- No circuito 2 adicione mais um Led.

Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

Servo Motor

Um servo motor é um equipamento eletromecânico que possui um encoder e um controlador acoplado (VIDAL, 2017). Esses sensores o tornam diferente de outros motores, suas funções são fornecer a velocidade e posicionamento final do motor, trabalhando com servomecanismo que usa realimentação de posição.

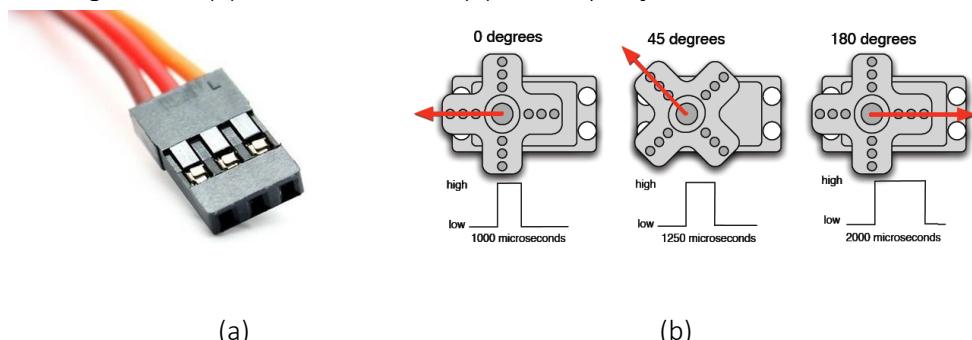
Figura 133: (a) Micro Servo 9g SG90 TowerPro e (b) Estrutura do Servo.



Fonte: (a) Imagem tirada de (Fazedores, 2018) e (b) Imagem tirada de (Pimenta, 2015).

Os Servos são acionados por meio de três fios, vermelho sendo a alimentação positiva, marrom GND e o laranja sinal que controla o PWM (que determina a posição final do motor, representado na figura 134-b).

Figura 134: (a) Fios de conexão e (b) PWM - posição do servo motor.



Fonte: (a) Imagem tirada de (Eletrogate, 2017) e (b) Imagem tirada de (Pimenta, 2015).

Praticando:

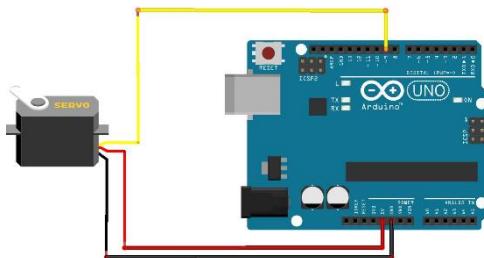
1- Controlando um servo motor.

Materiais necessários:

- 1 Servo Motor;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 135: Circuito.



Fonte: Imagem tirada de (UsinalInfo, 2016).

Código:

```
#include "Servo.h"
Servo meuservo;
int angulo = 0; // Ajusta o ângulo inicial do Servo

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

void loop() {
    for (angulo = 0; angulo < 180; angulo += 1) {
        meuservo.write(angulo); // Comando para ângulo específico
        delay(15);
    }
    delay(2000);
    for (angulo = 180; angulo >= 1; angulo -= 5) {
        meuservo.write(angulo);
        delay(5);
    }
    delay(2000);
}
```

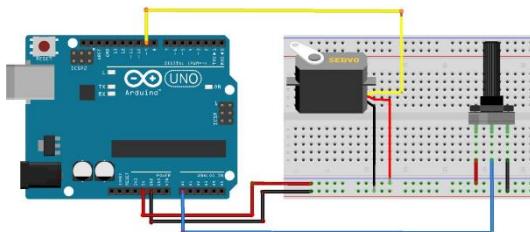
2- Controlando um servo motor com um potenciômetro.

Materiais necessários:

- 1 Servo Motor;
- 1 Arduino Uno;
- 1 Potenciômetro;
- 1 Protoboard;
- Jumpers diversos.

Círculo:

Figura 136: Círculo.



Fonte: Imagem tirada de (UsinaInfo, 2016).

Código:

```
#include Servo.h;
Servo meuservo;
int angulo = 0;
int potencio = A0;

void setup() {
meuservo.attach(9);
}
void loop() {
angulo = analogRead(potencio); // Faz a leitura do valor do potenciômetro
angulo = map(angulo, 0, 1023, 0, 179); // Associa o valor do potenciômetro ao valor do ângulo
meuservo.write(angulo); // Comando para posicionar o servo no ângulo especificado
delay(5);
}
```

DESAFIO

Em cada um dos circuitos acima realize esses testes:

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;
- No circuito 1, adicione um botão, que quando for apertado deve acionar o motor e quando apertado novamente deve retornar de onde voltou.

Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

Motor de Passo

Um motor de passo é um dispositivo eletromecânico que transforma os impulsos elétricos em movimentos discretos mecânicos (THOMSEN, 2013). Estes impulsos estão ligados com a direção em que o motor irá girar, sua frequência está ligada a velocidade de rotação e seu número com o comprimento da rotação.

Comumente é utilizado em projetos que precisam de precisão, pois sua posição é conhecida simplesmente através do controle dos impulsos de entrada, sem precisar de sistemas mais caros.

Figura 137: Motor de passo.



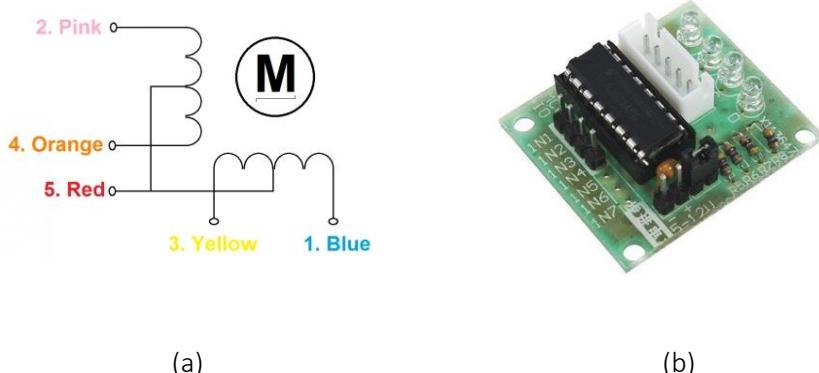
Fonte: Imagem tirada de (Master Walker, 2018).

Sua conexão conta com 4 fases (chamada também de bobinas ou enrolamentos) e 5 fios (4 fases + 1 comum), tendo um bom funcionamento quando aplicarmos os impulsos elétricos, conectando cada fio ao Arduino da seguinte forma:

- IN1 ao pino 8 do Arduino
- IN2 ao pino 9 do Arduino
- IN3 ao pino 10 do Arduino
- IN4 ao pino 11 do Arduino

Cada pino do Arduino irá acionar cada uma das fases do motor de passo, fazendo seu eixo girar por meio de um circuito Driver ULN2003 (que também se encontra na caixa de robótica), responsável por converter os sinais de pulso do controlador em movimento do motor.

Figura 138: (a) Funcionamento das fases e (b) Driver ULN2003.



Fonte: (a) Imagem tirada de (FilipeFlop, 2013) e (b) Imagem tirada de (FilipeFlop, 2020).

Praticando:

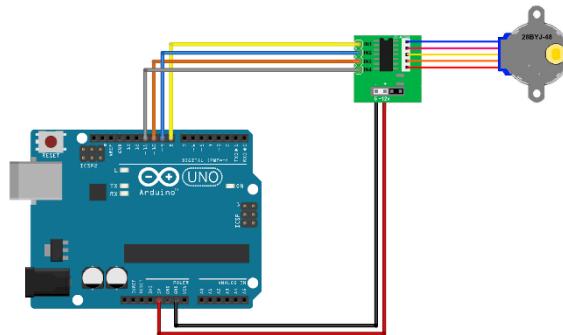
1- Controlando um motor de passo.

Materiais necessários:

- 1 Motor de Passo com Driver ULN2003;
- 1 Arduino Uno;
- Jumpers diversos.

Círculo:

Figura 139: Círculo.



Fonte: Imagem tirada de (FilipeFlop, 2013).

Código:

```
#include <Stepper.h>
const int stepsPerRevolution = 500;
//Utilizando as portas digitais 08 a 11 para conexão ao motor
Stepper myStepper(stepsPerRevolution, 8,10,9,11);

void setup() {
  // Velocidade inicial do motor (MAX 100)
  myStepper.setSpeed(60);
}

void loop() {
  for (int i = 0; i<=3; i++) { //Gira o eixo do motor no sentido horário - 90 graus
    myStepper.step(-512);
    delay(2000);
  }
  for (int i = 0; i<=2; i++) { //Gira o eixo do motor no sentido anti-horário - 120 graus
    myStepper.step(682);
    delay(2000);
  }
  for (int i = 10; i<=60; i=i+10) { //Gira o eixo do motor no sentido horário
    myStepper.setSpeed(i);
    myStepper.step(40*i);
  }
  delay(2000);
}
```

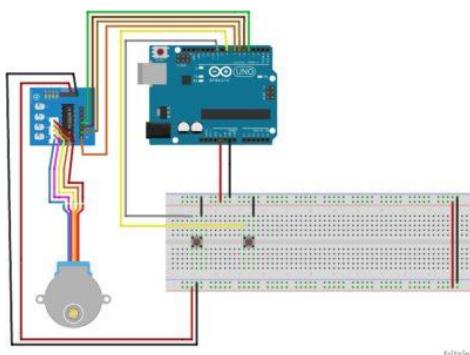
2- Controlando um motor de passo com botões.

Materiais necessários:

- 1 Motor de Passo com Driver ULN2003
- 2 push buttons
- 1 Arduino Uno
- 1 protoboard
- Jumpers diversos

Círculo:

Figura 140: Círculo.



Fonte: Imagem tirada de (Baú da Eletrônica, 2017).

Código:

```
#include "Stepper_28BYJ_48.h" // declaração biblioteca
int switch_1_pin = 8; //pino do primeiro botão
int switch_2_pin = 10; // pino do segundo botão
Stepper_28BYJ_48 stepper(7,6,5,4); // Declaração da instância do motor

void setup() {
    pinMode(switch_1_pin,INPUT_PULLUP);
    pinMode(switch_2_pin,INPUT_PULLUP);
}

void loop() {
    if (digitalRead(switch_1_pin) == LOW ) { // verifica se botão 1 foi pressionado
        stepper.step(-1); //gira motor sentido anti horário
        delay(10);
    }
    if ( digitalRead(switch_2_pin) == LOW) { // verifica se botão 1 foi pressionado
        stepper.step(1); // gira motor sentido horário
        delay(10);
    }
}
```

DESAFIO

Em cada um dos circuitos acima realize esses testes:

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;
- No circuito 2, adicione um botão para desligar e ligar o motor.

Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

Módulo Relé 1 Canal

É um componente eletromecânico; quando é utilizado cargas com valores de corrente ou tensão maiores de 40mA (máximo que a placa Arduino suporta) precisamos utilizar algum dispositivo que suporte tal carga para que não danifique a placa, sendo ele o módulo relé. Pode ser utilizado para acender uma lâmpada ou um motor.

Figura 141: Módulo Relé 1 canal.



Fonte: Imagem tirada de (Vida de silício, 2017).

Possui os seguintes pinos:

- Vcc - Pino para alimentação 5V;
- GND - Pino para Terra 0v;
- IN - Pino de comando do módulo. Esse é o pino responsável por ligar ou desligar o relé.

Praticando:

1- Piscando Led.

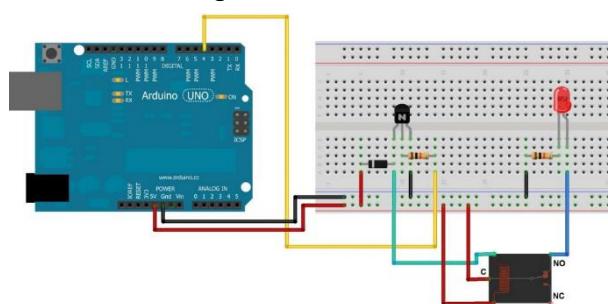
Materiais necessários:

- 1 Módulo Relé 1 canal;
- 1 LED;
- 1 diodo 1N4007;
- 2 resistores de 10K ohms;
- 1 Protoboard;
- 1 Transistor BC548;
- 1 Arduino Uno;

- Jumpers diversos.

Círculo:

Figura 142: Círculo.



Fonte: Imagem tirada de (Arduino e CIA, 2013).

Código:

```
int sinalparaorele = 4;

void setup(){
  //Define o pino como saída
  pinMode(sinalparaorele, OUTPUT);
}

void loop(){
  digitalWrite(sinalparaorele, HIGH); //Aciona o rele
  delay(5000);
  digitalWrite(sinalparaorele, LOW); //Desliga o rele
  delay(5000);
}
```

DESAFIO

No circuito acima realize esses testes:

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;
- Adicione outro LED.

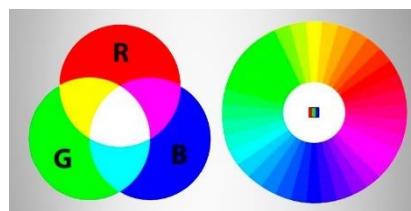
Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

OUTROS SENSORES E ELEMENTOS DA ROBÓTICA

LED RGB

Um LED RGB é uma junção de três LEDs em um, formado por um vermelho (R red), um verde (G green) e um azul (B blue). Podem ser controlados de maneira individual, conectando o terminal da cor diretamente a uma fonte de energia e deixando os outros desconectados, ou utilizar todos ao mesmo tempo, ajustando sua intensidade por meio de pulsos PWM, resultando em cores novas.

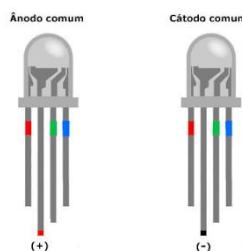
Figura 143: Formação de novas cores.



Fonte: Imagem tirada de (Baú da Eletrônica, 2019).

Composto por quatro terminais, sendo que três deles são conectados a cada LED de cor, e o quarto terminal é comum a todos que é classificado em dois tipos: anodo comum, pinos de cores vermelha, verde e azul são conectados ao terminal negativo (GND), e catodo comum, pinos de cores vermelha, verde e azul são conectados ao terminal positivo.

Figura 143: Anodo comum e catodo comum.



Fonte: Imagem tirada de (Canaleet, 2016).

Praticando:

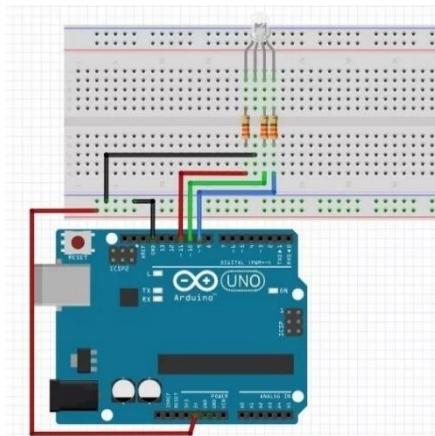
1- Acendendo LED.

Materiais necessários:

- 1 LED RGB;
- 3 Resistores de 330 ohms;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 144: Circuito.



Fonte: Imagem tirada de (Vida de Silício, 2018).

Código:

```
const int azul = 9;
const int verde = 10;
const int vermelho = 11;
String cor;

void setup() {
  Serial.begin(9600);
  pinMode(azul, OUTPUT);
  pinMode(verde, OUTPUT);
  pinMode(vermelho, OUTPUT);
}

//Funções responsáveis por executar o brilho selecionado
void vermelhoFuncao(){
  digitalWrite(azul, LOW);
  digitalWrite(verde, LOW);
  digitalWrite(vermelho, HIGH);
}

void azulFuncao(){
  digitalWrite(azul, HIGH);
  digitalWrite(verde, LOW);
  digitalWrite(vermelho, LOW);
}

void verdeFuncao(){
  digitalWrite(azul, LOW);
  digitalWrite(verde, HIGH);
  digitalWrite(vermelho, LOW);
}

void loop() {
  if(Serial.available()){
    cor = Serial.readString();
    Serial.println(cor);
  }
}
```

```

if(cor == "Vermelho"){
    vermelhoFuncao();
}

if(cor == "Azul"){
    azulFuncao();
}

if(cor == "Verde"){
    verdeFuncao();
}
}

```

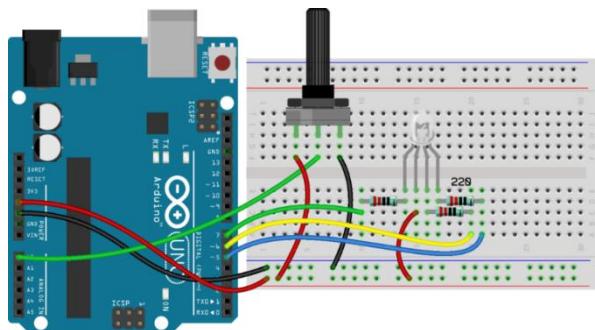
2- Trocando as cores do LED.

Materiais necessários:

- 1 LED RGB;
- 1 Potenciômetro;
- 3 Resistores de 330 ohms;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 145: Circuito.



Fonte: Imagem tirada de (FilipeFlop, 2018).

Código:

```

int led_R = 7;
int led_G = 6;
int led_B = 5;
int pot = A0;
int valorPot;

void setup() {
    pinMode(led_R, OUTPUT);
    pinMode(led_G, OUTPUT);
    pinMode(led_B, OUTPUT);
    apagaLed();
}

void loop() {
    valorPot = analogRead(pot); // lê o valor do potenciômetro (de 0 a 1023)
}

```

```

if(valorPot >= 0 && valorPot <= 256) { // menor que 256 apaga o LED
    apagaLed();
}
if(valorPot > 256 && valorPot <= 512) { // entre 256 e 512, acende vermelho
    acendeVermelho();
}
if(valorPot > 512 && valorPot <= 768) { // entre 512 e 768, acende verde
    acendeVerde();
}
if(valorPot > 768 && valorPot <= 1023){ // entre 768 e 1023, acende azul
    acendeAzul();
}

void acendeVermelho() {
    digitalWrite(led_R, LOW);
    digitalWrite(led_G, HIGH);
    digitalWrite(led_B, HIGH);
}

void acendeVerde() {
    digitalWrite(led_R, HIGH);
    digitalWrite(led_G, LOW);
    digitalWrite(led_B, HIGH);
}

void acendeAzul(){
    digitalWrite(led_R, HIGH);
    digitalWrite(led_G, HIGH);
    digitalWrite(led_B, LOW);
}

void apagaLed() {
    digitalWrite(led_R, HIGH);
    digitalWrite(led_G, HIGH);
    digitalWrite(led_B, HIGH);
}

```

DESAFIO

Nos circuitos acima realize esses testes:

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;
- No 1º circuito adicione novas cores;
- No 2º circuito adicione um potenciômetro para cada cor, assim que você girar você vai controlar a intensidade da cor que aquele potenciômetro representa.

Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

Sensor NTC 10K

Assim como o sensor NTC citado no capítulo anterior, sua resistência diminui com o aumento da temperatura. Seu diferencial é ser à prova d'água, é extremamente utilizado em projetos que se faz necessária a medição de temperatura da água por imersão.

Figura 146: Sensor NTC 10K.



Fonte: Imagem tirada de (AutoCore, 2020).

Praticando:

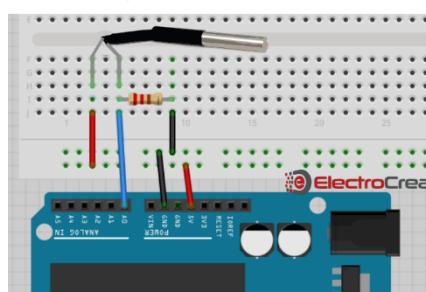
1- Apresentando temperaturas em Celsius e Kelvin.

Materiais necessários:

- 1 Sensor NTC 10K;
- 1 Resistor de 10K ohms;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círculo:

Figura 147: Circuito.



Fonte: Imagem tirada de (FilipeFlop, 2018).

Código:

```
#include <math.h> <br>

void setup () {
Serial.begin (115200); // Lembre-se de definir seu monitor serial como 115200
}

Thermister double (int RawADC) {
Temp double;
Temp = log (((10240000 / RawADC) - 10000));
Temp = 1 / (0,001129148 + (0,000234125 + (0,0000000876741 * Temp * Temp)) * Temp);
Temp = Temp - 273.15; // Converter de Kelvin em Celsius
// Para converter Celsius em Farenheith, escreva nesta linha: Temp = (Temp * 9.0) / 5.0 + 32.0;
return Temp;
}

void loop () {
int val; // Cria uma variável inteira
double temp; // variável de temperatura = temp
val = analogRead (0); // Leia o valor do pino analógico 0 e mantenha-o como val
temp = Thermister (val); // Realiza a conversão do valor analógico em graus Celsius
Serial.println (temp); // Grava a temperatura no monitor serial
delay (1000);
}
```

DESAFIO

No circuito acima realize esses testes:

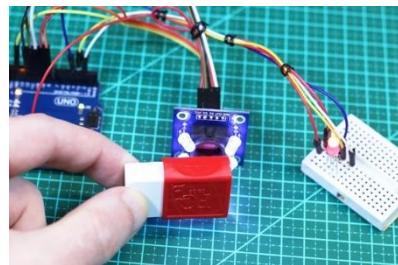
- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard.

Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

Sensor RGB

O sensor RGB identifica a cor do objeto que é lido, isso acontece por ter quatro fototransistores, um para cada cor (vermelho, verde e azul) e outro neutro que é responsável pelo sensor detectar outros tons de cores partindo dos que já estão definidos (RGB). Seu funcionamento ocorre partir da incidência de luz na base do transistor, conduzindo a corrente de acordo com a cor que está sobre ele. Existem três tipos de sensores RGB: sensor RGB TCS230, TCS3200 e TCS34725.

Figura 148: Sensor RGB em funcionamento.



Fonte: Imagem tirada de (FilipeFlop, 2016).

O sensor RGB TCS230 é composto por 64 fotodiodos, 16 tem filtros para vermelha, 16 filtros para a verde, 16 para a azul e 16 não tem filtro algum. Composto por 8 pinos: S0, S1, S2, S3 (utilizado para o controle do sensor), pino OUT (responsável pelo envio das informações coletadas), pino OE (Output Enable), deve ser ligado ao GND (outro pino), e o pino VCC.

Figura 149: Sensor RGB TCS230.



Fonte: Imagem tirada de (SECCI, 2019).

O sensor TCS3200, é uma atualização do sensor TCS230. Utiliza 10 pinos ao invés de 8 pinos, ele tem duas fileiras de 5 pinos, onde encontramos os pinos de controle (S0, S1, S2, S3), saída (OUT), controle do LED (LED) e alimentação (VCC e GND).

Figura 149: Sensor RGB TCS3200.



Fonte: Imagem tirada de (SECCI, 2019).

O sensor TCS34725 possui sensores de luz RGB que em conjunto com o filtro IR que minimizam a influência do espectro IR. Possui um regulador de tensão 3.3V, alimentado com 3-5VDC, um LED neutro afim de iluminar melhor o objeto a ser lido e certificar que a resposta será correta, que pode ser facilmente ligado e desligado através de um sinal digital.

Figura 150: Sensor RGB TCS34725.



Fonte: Imagem tirada de (SECCI, 2019).

Praticando:

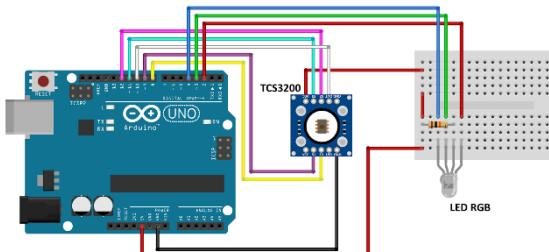
1- Acendendo um LED com a cor do objeto lido pelo sensor RGB.

Materiais necessários:

- 1 Sensor RGB TCS3200;
- 1 LED RGB;
- 1 Resistor de 10K ohms;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círculo:

Figura 151: Circuito.



Fonte: Imagem tirada de (FilipeFlop, 2016).

Código:

```
const int s0 = 8;
const int s1 = 9;
const int s2 = 12;
const int s3 = 11;
const int out = 10;
int pinoledverm = 2;
int pinoledverd = 3;
int pinoledazul = 4;
int red = 0;
int green = 0;
int blue = 0;

void setup(){
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
```

```

pinMode(s3, OUTPUT);
pinMode(out, INPUT);
pinMode(pinoledverm, OUTPUT);
pinMode(pinoledverd, OUTPUT);
pinMode(pinoledazul, OUTPUT);
Serial.begin(9600);
digitalWrite(s0, HIGH);
digitalWrite(s1, LOW);
}

void loop(){
//Detecta a cor
color();
//Mostra valores no serial monitor
Serial.print("Vermelho :");
Serial.print(red, DEC);
Serial.print(" Verde : ");
Serial.print(green, DEC);
Serial.print(" Azul : ");
Serial.print(blue, DEC);
Serial.println();
//Verifica se a cor vermelha foi detectada
if (red < blue && red < green && red < 100) {
  Serial.println("Vermelho");
  digitalWrite(pinoledverm, LOW); //Acende o led vermelho
  digitalWrite(pinoledverd, HIGH);
  digitalWrite(pinoledazul, HIGH);
}
//Verifica se a cor azul foi detectada
else if (blue < red && blue < green && blue < 1000){
  Serial.println("Azul");
  digitalWrite(pinoledverm, HIGH);
  digitalWrite(pinoledverd, HIGH);
  digitalWrite(pinoledazul, LOW); //Acende o led azul
}
//Verifica se a cor verde foi detectada
else if (green < red && green < blue){
  Serial.println("Verde");
  digitalWrite(pinoledverm, HIGH);
  digitalWrite(pinoledverd, LOW); //Acende o led verde
  digitalWrite(pinoledazul, HIGH);
}
Serial.println();
//Delay para apagar os leds e reiniciar o processo
delay(50);
digitalWrite(pinoledverm, HIGH);
digitalWrite(pinoledverd, HIGH);
digitalWrite(pinoledazul, HIGH);
}

void color(){
//Rotina que le o valor das cores

```

```

digitalWrite(s2, LOW);
digitalWrite(s3, LOW);
//count OUT, pRed, RED
red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
digitalWrite(s3, HIGH);
//count OUT, pBLUE, BLUE
blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
digitalWrite(s2, HIGH);
//count OUT, pGreen, GREEN
green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
}

```

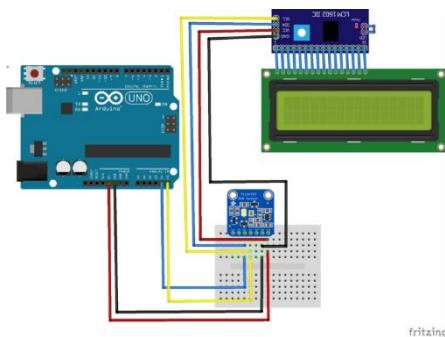
2- Apresentando em um Display LCD a cor do objeto lido pelo sensor RGB.

Materiais necessários:

- 1 Sensor RGB TCS3200;
- 1 Display LCD;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Circuito:

Figura 152: Circuito.



Fonte: Imagem tirada de (Baú da Eletrônica, 2017).

Código: Para o funcionamento do código instale essas bibliotecas Adafruit_TCS34725 e New Liquid Crystal (clicando em cima do nome de cada).

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "Adafruit_TCS34725.h"
LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7,3, POSITIVE);
#define BACKLIGHT_PIN 13
// instância do sensor RGB
Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_700MS,
TCS34725_GAIN_1X);

void setup() {
pinMode ( BACKLIGHT_PIN, OUTPUT );
digitalWrite ( BACKLIGHT_PIN, HIGH );
lcd.begin(16,2);
lcd.home ();

```

```

lcd.print("Identificação de cores");
lcd.setCursor ( 0, 1 );
lcd.print ("Testando display");
delay ( 1000 );
lcd.clear();
if (tcs.begin()) { // se o sensor foi detectado
    lcd.setCursor ( 0, 0 );
    lcd.println("Sensor ok      ");
} else { // se sensor não encontrado
    lcd.setCursor ( 0, 0 );
    lcd.println("Nao ha sensor");
    lcd.setCursor ( 0, 1 );
    lcd.print("Verifique!");
    while (1);
}
}

void loop(){
    // instância para variáveis do sensor
    uint16_t r, g, b, c, colorTemp, lux;
    // leitura de dados de tons de cores e luminosidade
    tcs.getRawData(&r, &g, &b, &c);
    // cálculo dos níveis de cores
    colorTemp = tcs.calculateColorTemperature(r, g, b);
    // cálculo de nível de luminosidade
    lux = tcs.calculateLux(r, g, b);
    if(c < 5000){ // se claridade for menor que 5000
        if (r>b && r>g){ // tom vermelho maiores que azul e verde
            lcd.setCursor(0,1);
            lcd.print("Vermelho      ");
            delay(1000);
        }
        else if(g>r && g>b){ // tom verde maiores que azul e vermelho
            lcd.setCursor(0,1);
            lcd.print("Verde      ");
            delay(1000);
        }
        else if(b>r && b>g){ // tom azul maiores que vermelho e verde
            lcd.setCursor(0,1);
            lcd.print("Azul      ");
            delay(1000);
        }
    }
}
}
}

```

DESAFIO

Nos circuitos acima realize esses testes:

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;
- Inclua mais cores para leitura;

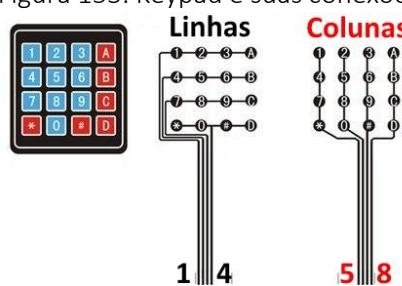
- Em ambos, troque para o sensor TCS230, no 1º também troque para o sensor TCS34725 e no 2º troque para o sensor TCS3200;
- Depois de testar os três sensores nos dois experimentos, faça anotações sobre os resultados obtidos e discutam sobre qual foi o mais eficiente.

Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

Keypad

Este componente é composto por 16 teclas (4 linhas e 4 colunas) e embaixo de cada tecla há um interruptor de membrana, que é conectado aos outros interruptores da mesma linha, esses são conectados as colunas (também conectadas). Sua função é facilitar a entrada de dados.

Figura 153: Keypad e suas conexões.

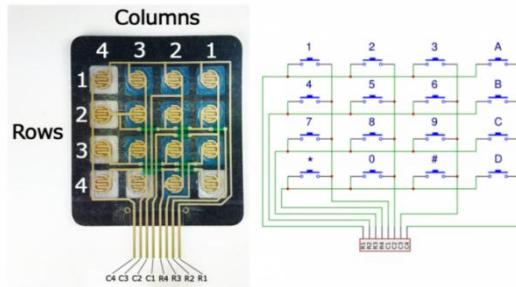


Fonte: Imagem tirada de (Baú da Eletrônica, 2017).

Para identificar o botão qual foi pressionado, o Arduino segue os quatro seguintes passos:

1. Configura as colunas como entradas em nível lógico alto, ativando o resistor de *pull-up*, e todas as linhas como nível lógico baixo;
2. Caso um botão seja pressionado, a coluna deste botão será transformada em nível lógico baixo;
3. Assim as linhas ficam com nível lógico alto e as colunas com nível lógico baixo;
4. Para finalizar, a linha do botão que anteriormente foi pressionado assumirá o nível lógico baixo, sendo identificado qual botão foi pressionado.

Figura 154: Conexões Keypad.



Fonte: Imagem tirada de (RoboCore, 2019).

Praticando:

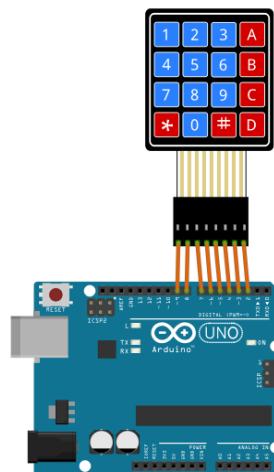
1- Exibindo no monitor serial as teclas pressionadas.

Materiais necessários:

- 1 Keypad;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 155: Circuito.



Fonte: Imagem tirada de (Baú da Eletrônica, 2017).

Código: Para o funcionamento do código instale essa biblioteca ([Download biblioteca Keypad](#)).

```
#include <Keypad.h>
const byte LINHAS = 4;
const byte COLUNAS = 4;
const char TECLAS_MATRIZ[LINHAS][COLUNAS] = { // Matriz de caracteres (mapeamento do teclado)
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};
const byte PINOS_LINHAS[LINHAS] = {9, 8, 7, 6}; // Pinos de conexão com as linhas do teclado
const byte PINOS_COLUNAS[COLUNAS] = {5, 4, 3, 2}; // Pinos de conexão com as colunas
```

```

Keypad teclado_personalizado = Keypad(makeKeymap(TECLAS_MATRIZ), PINOS_LINHAS,
PINOS_COLUNAS, LINHAS, COLUNAS); // Inicia teclado

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

void loop() {
  char leitura_teclas = teclado_personalizado.getKey(); // Atribui a variavel a leitura do teclado
  if (leitura_teclas) { // Se alguma tecla foi pressionada
    Serial.println(leitura_teclas); // Imprime a tecla pressionada na porta serial
  }
}

```

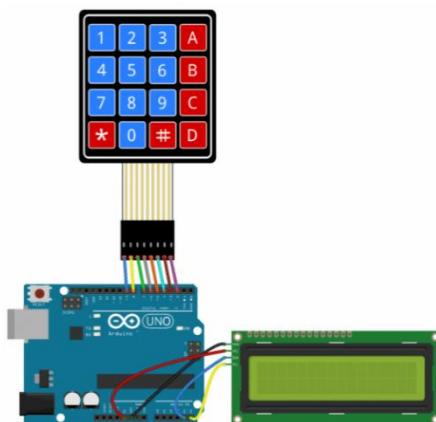
2- Exibindo no Display LCD as teclas pressionadas.

Materiais necessários:

- 1 Keypad;
- Display LCD;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 156: Circuito.



Fonte: Imagem tirada de (Circuit Basics, 2017).

Código:

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {9, 8, 7, 6};
byte colPins[COLS] = {5, 4, 3, 2};

```

```

Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
LiquidCrystal_I2C lcd(0x21, 16, 2);

void setup(){
  lcd.backlight();
  lcd.init();
}

void loop(){
  char customKey = customKeypad.getKey();
  if (customKey){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(customKey);
  }
}

```

DESAFIO

Nos circuitos acima realize esse teste:

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;

Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

Display 7 Segmentos

É um componente eletrônico que é utilizado como saída de informações de forma visual. É composto por apenas um dígito, mas cada segmento pode ser acionado de forma independente, formando números e letras. Os segmentos são identificados com letras de A até G e mais DP para o ponto decimal.

Imagem 157: (a) Segmentos do display 7 segmentos e (b) Pinos de conexão dos segmentos.



(a)

(b)

Fontes: (a) Imagem tirada de (Eletrogate, 2019) e (b) Imagem tirada de (Squids, 2019).

Podem ser anodo comum ou catodo comum. Anodo quando os segmentos estão conectados ao Vcc, acionados por LOW ou 0. Catodo quando todos os segmentos são conectados ao terra, acionado por HIGH ou 1.

Praticando:

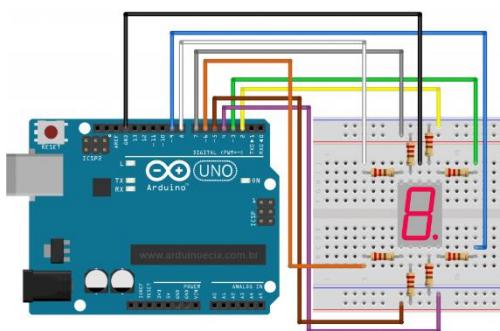
1- Acionando o display 7 segmentos.

Materiais necessários:

- 1 Display 7 segmentos;
- 8 Resistores 220 ohms;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 158: Circuito.



Fonte: Imagem tirada de (Arduino e CIA, 2013).

Código:

```
byte seven_seg_digits[16][7] = { { 1,1,1,1,1,1,0 }, // = Digito 0
{ 0,1,1,0,0,0,0 }, // = Digito 1
{ 1,1,0,1,1,0,1 }, // = Digito 2
{ 1,1,1,1,0,0,1 }, // = Digito 3
{ 0,1,1,0,0,1,1 }, // = Digito 4
{ 1,0,1,1,0,1,1 }, // = Digito 5
{ 1,0,1,1,1,1,1 }, // = Digito 6
{ 1,1,1,0,0,0,0 }, // = Digito 7
{ 1,1,1,1,1,1,1 }, // = Digito 8
{ 1,1,1,0,0,1,1 }, // = Digito 9
{ 1,1,1,0,1,1,1 }, // = Digito A
{ 0,0,1,1,1,1,1 }, // = Digito B
{ 1,0,0,1,1,1,0 }, // = Digito C
{ 0,1,1,1,1,0,1 }, // = Digito D
{ 1,0,0,1,1,1,1 }, // = Digito E
{ 1,0,0,0,1,1,1 } // = Digito F
};
```

```

void setup() {
    pinMode(2, OUTPUT); //Pino 2 do Arduino ligado ao segmento A
    pinMode(3, OUTPUT); //Pino 3 do Arduino ligado ao segmento B
    pinMode(4, OUTPUT); //Pino 4 do Arduino ligado ao segmento C
    pinMode(5, OUTPUT); //Pino 5 do Arduino ligado ao segmento D
    pinMode(6, OUTPUT); //Pino 6 do Arduino ligado ao segmento E
    pinMode(7, OUTPUT); //Pino 7 do Arduino ligado ao segmento F
    pinMode(8, OUTPUT); //Pino 8 do Arduino ligado ao segmento G
    pinMode(9, OUTPUT); //Pino 9 do Arduino ligado ao segmento PONTO
    writePonto(0); // Inicia com o ponto desligado
}
void writePonto(byte dot) { //Funcao que aciona o ponto no display
    digitalWrite(9, dot);
}
void sevenSegWrite(byte digit){ //Funcao que aciona o display
    byte pin = 2;
    //Percorre o array ligando os segmentos correspondentes ao digito
    for (byte segCount = 0; segCount < 7; ++segCount){
        digitalWrite(pin, seven_seg_digits[digit][segCount]);
        ++pin;
    }
    writePonto(1); //Liga o ponto
    delay(100);
    writePonto(0); //Desliga o ponto
}

void loop() {
    //Contador de 0 a 15, ligando os segmentos correspondentes
    //0 a 9 = liga os segmentos correspondentes aos numeros
    //10 a 15 = Forma as letras A,B,C,D,E,F
    for (byte count = 0; count < 16; count++){
        delay(500);
        sevenSegWrite(count);
    }
    delay(4000);
}

```

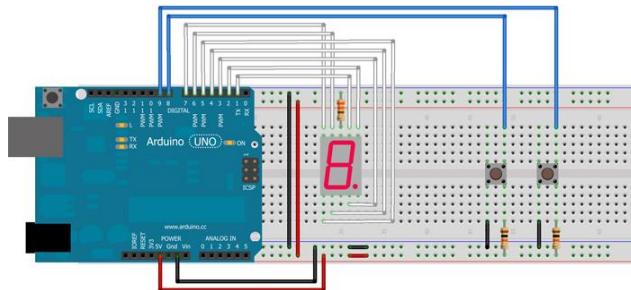
2- Crescendo ou decrescendo com botões (ao clicar em um botão aumentará o número, ou clicar no outro o número irá decrescer).

Materiais necessários:

- 1 Display 7 segmentos;
- 2 Push-buttons;
- 8 Resistores 220 ohms;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Circuito:

Figura 159: Circuito.



Fonte: Imagem tirada de (Arduino, 2016).

Código:

```
#define CONTAR 8 // pino onde será conectado a tecla de contagem
#define MODO 9 // pino onde será conectado a tecla de modo
byte tabela_7_seg [16] = {
B0111110, // = Digito 0
B00001100, // = Digito 1
B10110110, // = Digito 2
B10011110, // = Digito 3
B11001100, // = Digito 4
B11011010, // = Digito 5
B11111010, // = Digito 6
B00001110, // = Digito 7
B11111110, // = Digito 8
B11001110, // = Digito 9
B11101110, // = Digito A
B1111000, // = Digito B
B01110010, // = Digito C
B10111100, // = Digito D
B11110010, // = Digito E
B11100010 // = Digito F
};
int contagem = 9, inc_dec =1;

void setup() {
  DDRD = B1111110; //Configura os pinos digitais de 1 até 7 como saída digital
  pinMode(CONTAR, INPUT);
  pinMode(MODO, INPUT);
}

void loop() {
  // Lê o botão de Contagem
  if (digitalRead(CONTAR)== LOW)  {
    contagem += inc_dec;
    //Reseta a contagem em caso de estouro
    if (contagem < 0)  {
      contagem = 15;
    }
    if (contagem > 15)  {
      contagem = 0;
    }
    //Aguarda a tecla de contagem ser solta
  }
}
```

```

while(digitalRead(CONTAR)== LOW) {
}
}
// Lê o botão que seleciona MODO
if(digitalRead(MODO)== LOW) {
    //Inverte o modo de operação do contador
    inc_dec = inc_dec * (-1);
    //Aguarda a tecla de MODO ser solta
    while(digitalRead(MODO)== LOW){
    }
}
PORTD = tabela_7_seg[contagem]; //Busca o valor da contagem na matriz
delay(50);
}

```

DESAFIO

Nos circuitos acima realize esses testes:

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;
- No circuito 1 adicione outro display 7 segmentos, um apresentando apenas números e outro apenas letras;
- No circuito 2 adicione outro botão para apagar tudo o que estiver sendo apresentado no display 7 segmentos.

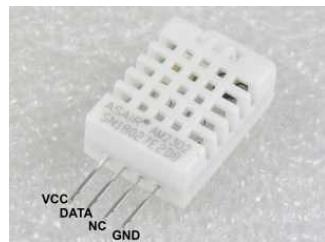
Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

DHT11 e DHT22

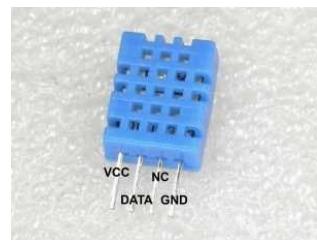
O sensor DHT11 e o sensor DHT22 são sensores que tem como função medir temperatura e umidade, utilizando um termistor e um sensor capacitivo. Seu tempo de intervalo para que não ocorra problema entre as leituras é de um segundo (SHIGUEMORI, 2019).

Seus pinos são formados por GND, DATA, VCC (no DHT11 vai de 3,5 até 5V e no DHT22 vai de 3,3 até 5V) e NC.

Imagen 160: (a) Pinos de conexão DHT22 e (b) Pinos de conexão DHT11.



(a)



(b)

Fonte: (a) Imagem tirada de (Eletrogate, 2019). e (b) Imagem tirada de (Eletrogate, 2019).

Tabela 17: Comparação entre o sensor DHT11 e DHT22.

	DHT11	DHT22
Faixa de umidade	20 a 80 %	0 a 100 %
Precisão na umidade	$\pm 5\%$ RH	$\pm 2\%$ RH
Resolução de umidade	5 % RH	0,1 % RH
Faixa de temperatura	0 a 50 °C	-40 a 80 °C
Precisão na temperatura	$\pm 2\%$ °C	$\pm 1\%$ °C
Resolução na temperatura	2 °C	0,1 °C
Tempo de resposta	2 segundos	< 5 segundos
Alimentação	3,5 V a 5 V	de 3,3V a 5 V
Consumo máximo de corrente	2,5 mA	0,5 mA

Praticando:

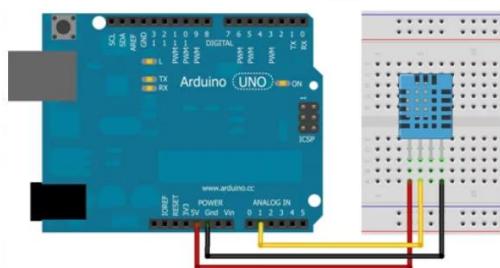
1- Medindo temperatura e umidade com DHT11 e apresentando no monitor serial.

Materiais necessários:

- 1 Sensor DHT11;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 161: Circuito.



Fonte: Imagem tirada de (Vida de Silício, 2017).

Código: Para o funcionamento desse código deve ter essa biblioteca instalada, clique aqui.

```
#include "DHT.h"
#define DHTPIN A1 // pino que estamos conectados
#define DHTTYPE DHT11 // DHT 11
DHT dht(DHTPIN, DHTTYPE);
void setup(){
    Serial.begin(9600);
    Serial.println("DHTxx test!");
    dht.begin();
}
void loop(){
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    // testa se retorno é valido, caso contrário algo está errado.
    if (isnan(t) || isnan(h)) {
        Serial.println("Failed to read from DHT");
    }
    else{
        Serial.print("Umidade: ");
        Serial.print(h);
        Serial.print(" %t");
        Serial.print("Temperatura: ");
        Serial.print(t);
        Serial.println(" *C");
    }
}
```

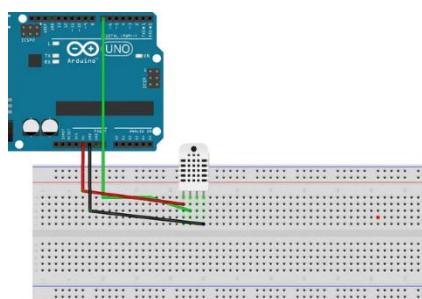
2- Medindo temperatura e umidade com DHT22 e apresentando no monitor serial.

Materiais necessários:

- 1 Sensor DHT22;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Circuito:

Figura 162: Circuito.



Fonte: Imagem tirada de (Arduino, 2016).

Código:

```
#include <DHT.h>
#define DHTPIN 7 // what pin we're connected to
#define DHTTYPE DHT22 // DHT 22 (AM2302)
DHT dht(DHTPIN, DHTTYPE); /// Initialize DHT sensor for normal 16mhz Arduino
int chk;
float hum; //Stores humidity value
float temp; //Stores temperature value

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

void loop() {
  delay(2000);
  //Read data and store it to variables hum and temp
  hum = dht.readHumidity();
  temp= dht.readTemperature();
  //Print temp and humidity values to serial monitor
  Serial.print("Humidity: ");
  Serial.print(hum);
  Serial.print(" %, Temp: ");
  Serial.print(temp);
  Serial.println(" Celsius");
  delay(10000); //Delay 2 sec.
}
```

DESAFIO

Nos circuitos acima realize esses testes:

- Monte o circuito de uma maneira diferente usando outros furos e posições na protoboard;
- Monte um novo circuito com os dois sensores (DHT11 e DHT22) e apresente os valores dos dois sensores, comparando-os.

Anote sobre os experimentos (resultados, dúvidas, curiosidades etc.).

EXPERIMENTOS AVANÇADOS

A partir desse capítulo, o código deve ser pensado, para treinar os conceitos de programação.

Semáforo comum

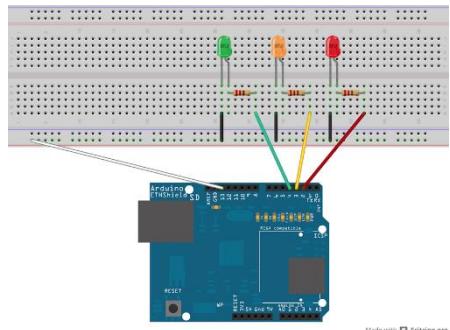
Neste experimento será simulado um sinal de trânsito na utilização de LEDs coloridos.

Materiais necessários:

- 1 LED verde;
- 1 LED vermelho;
- 1 LED amarelo;
- 3 Resistores 220 ohms;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Círcuito:

Figura 163: Circuito.



Fonte: Imagem tirada de (PIMENTA, 2014).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Semáforo interativo

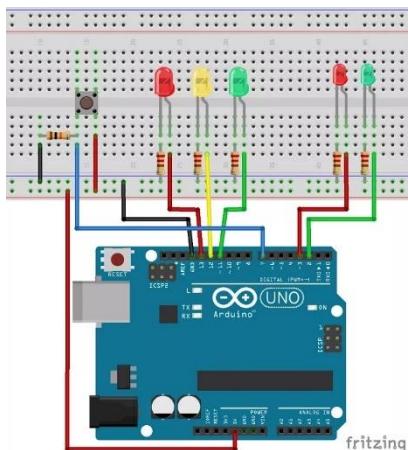
Este experimento consiste em um semáforo que possui botão para pedestre, quando apertado o sinal para carros deve passar pelo amarelo e ficar vermelho e o sinal de pedestres verde, para que o pedestre possa atravessar a rua.

Materiais necessários:

- 2 LEDs verde;
- 2 LEDs vermelho;
- 1 LED amarelo;
- 1 Push-button;
- 5 Resistores 220 ohms e 1 Resistor 10K ohms;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Circuito:

Figura 164: Circuito.



Fonte: Imagem tirada de (Eletrônica para todos, 2017).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Sequencial de LEDs

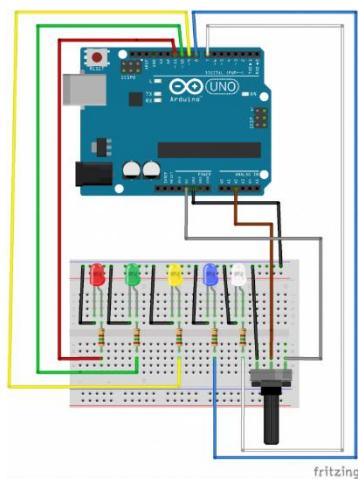
Neste experimento o sequencial de LED será incrementado com um potenciômetro, que tem como função variar a velocidade da iluminação dos LEDs.

Materiais necessários:

- 5 LEDs;
- 1 Potenciômetro;
- 5 Resistores 220 ohms;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Circuito:

Figura 165: Circuito.



Fonte: Imagem tirada de (Master Walker, 2019).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

LED RGB controlado por botões

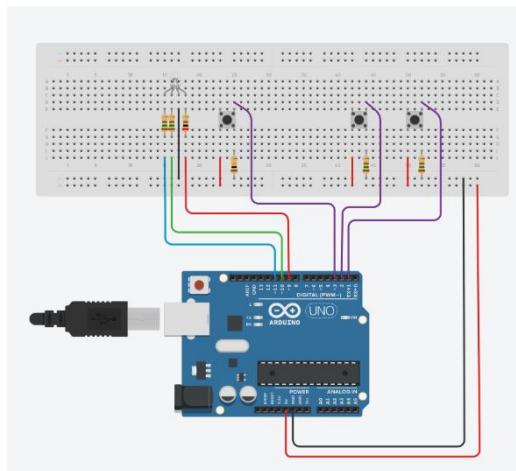
Neste experimento deve-se controlar cada uma das cores de um LED RGB utilizando botões, composto por um botão seletor de cor e outros dois ajustando a intensidade e o brilho do Led.

Materiais necessários:

- 1 LED RGB;
- 3 Push-buttons;
- 1 Resistor de 150 ohms, 2 resistores de 100 ohms e 3 resistores de 10K ohms;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Círculo:

Figura 166: Círculo.



Fonte: Imagem tirada de (Squids, 2017).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

LED RGB controlado por botões e potenciômetro

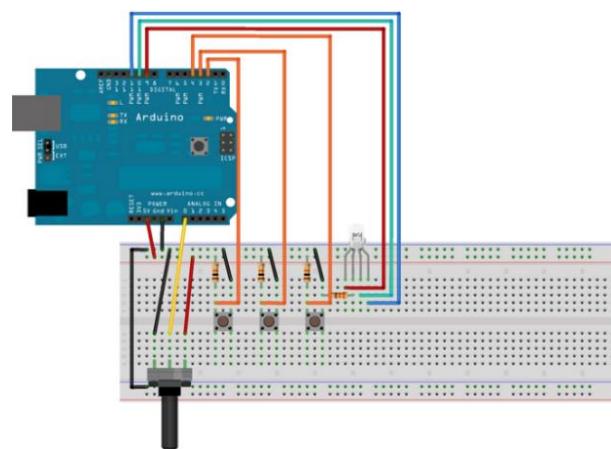
Neste experimento será utilizado um potenciômetro para controlar a intensidade das cores de um LED RGB, selecionando a cor através do botão.

Materiais necessários:

- 1 LED RGB;
- 3 Push-buttons;
- 1 Resistor de 330 ohms e 3 resistores de 10K ohms;
- 1 Potenciômetro;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Círcuito:

Figura 167: Círcuito.



Fonte: Imagem tirada de (Laboratório de Garagem, 2017).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Criando cores com LED RGB utilizando Monitor Serial

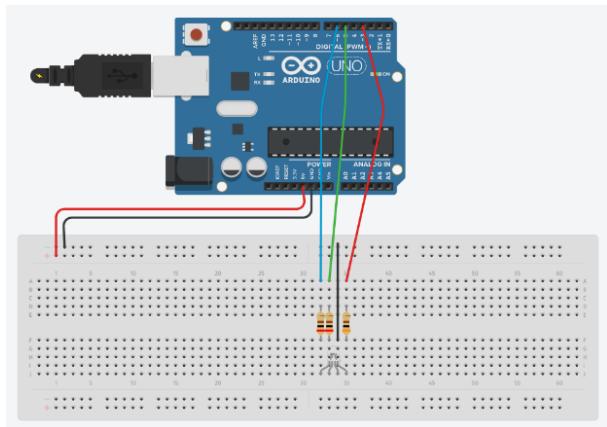
Neste experimento o Led RGB deve ser controlado por meio dos comandos que devem ser enviados via Monitor Serial. Ex: Caso a cor desejada for rosa, pesquise o RGB dessa cor e insira da seguinte forma por meio do Monitor Serial: r252, g15, b192. Combinando os valores de cada cor podemos criar até 16,7 milhões de cores diferentes.

Materiais necessários:

- 1 LED RGB;
- 3 Resistores de 330 ohms;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Circuito:

Figura 168: Circuito.



Fonte: Imagem tirada de (Squids, 2017).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Sensor de luz LDR com saída em LEDs

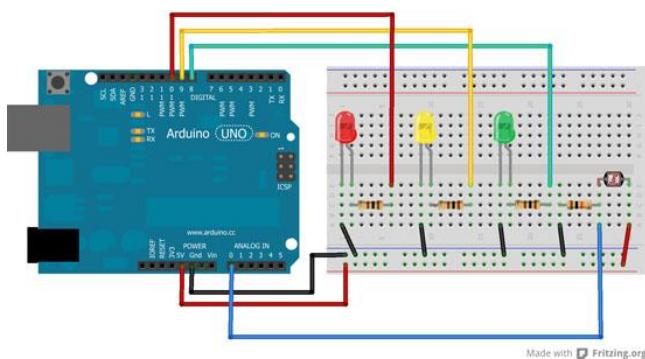
Este experimento faz a leitura da iluminação por meio do LDR e exibe o resultado nos LEDs, iluminação alta = LED verde acenderá, iluminação mediana = LED amarelo acenderá e iluminação baixa = LED vermelho acenderá. O resultado também será apresentado no monitor serial.

Materiais necessários:

- 1 LED Vermelho;
- 1 LED Verde;
- 1 LED Amarelo;
- 3 Resistores de 100 ohms e 1 Resistor de 10K ohms;
- 1 Sensor LDR;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Circuito:

Figura 169: Circuito.



Fonte: Imagem tirada de (Como fazer as coisas, 2013).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Sensor de luz LDR com sinalizador de LEDs e Buzzer

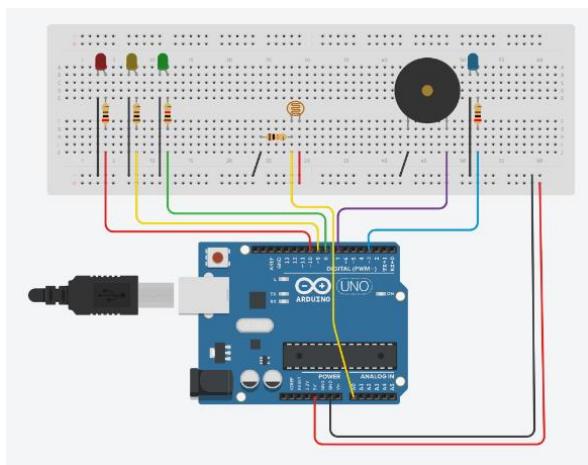
Assim como no experimento anterior ele irá alternar entre os LEDs de acordo com a iluminação do ambiente. O diferencial desse experimento é o buzzer e o LED que serão acionados quando houver pouca luz e desativado quando tiver mais luz.

Materiais necessários:

- 1 LED Vermelho;
- 1 LED Verde;
- 1 LED Amarelo;
- 1 LED Azul;
- 4 Resistores de 100 ohms e 1 Resistor de 10K ohms;
- 1 Buzzer;
- 1 Sensor LDR;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círculo:

Figura 170: Círculo.



Fonte: Imagem tirada de (Squids, 2017).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Jogo das Palavras Embaralhadas

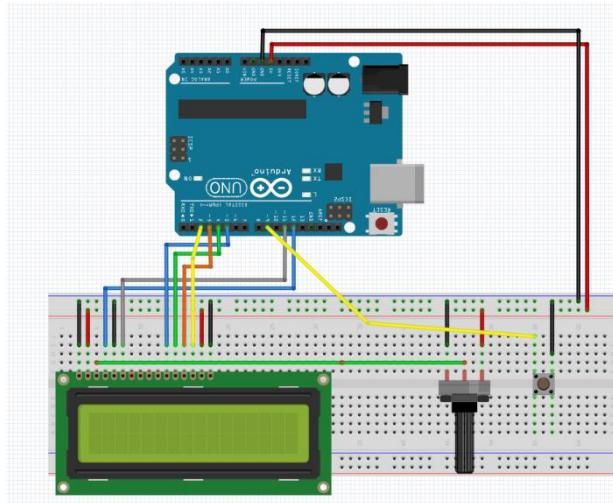
Neste experimento o Arduino embaralha as palavras digitadas por meio do monitor serial para que outra pessoa tente adivinhar a palavra correta. Ao clicar no botão a palavra deve ser desembaralhada.

Materiais necessários:

- 1 Display LCD;
- 1 Push-button;
- 1 Potenciômetro;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 171: Circuito.



Fonte: Imagem tirada de (Squids, 2018).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Alarme de temperatura com Buzzer

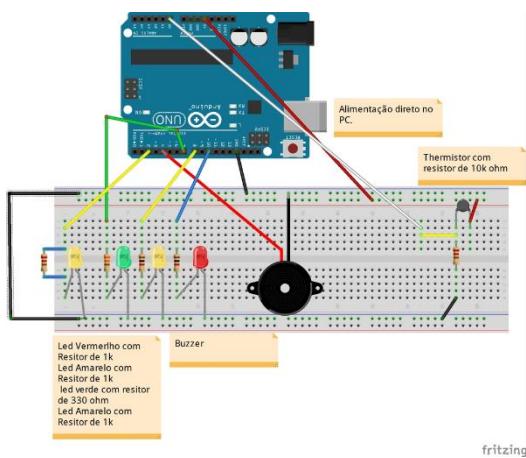
Neste experimento o Led verde indica que a temperatura está no padrão, o vermelho quando está muito frio ou muito quente e os LEDs amarelos indicam que a temperatura está no limite (quando estiver próximo ao limite máximo acende o LED amarelo próximo do vermelho, caso esteja próximo do valor mínimo, acenderá o LED amarelo perto do Led verde). O buzzer deve ser ativado juntamente com o Led vermelho.

Materiais necessários:

- 1 Buzzer;
- 1 LED Vermelho;
- 1 LED Verde;
- 2 LEDs Amarelo;
- 1 Resistor de 10K ohms e 4 Resistores 330 ohms;
- 1 Sensor NTC;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Circuito:

Figura 172: Circuito.



Fonte: Imagem tirada de (Laboratório de Garagem, 2014).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Piano com buzzer

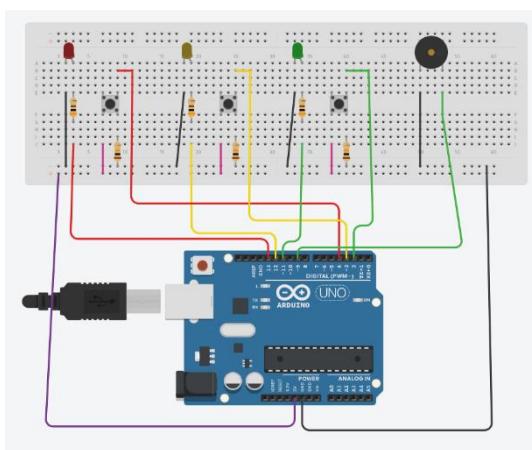
Ao pressionar o botão deve ser ativado o LED correspondente a ele e deve tocar uma nota musical.

Materiais necessários:

- 1 Buzzer;
- 1 LED Vermelho;
- 1 LED Verde;
- 1 LED Amarelo;
- 3 Resistor de 10K ohms e 3 Resistores 330 ohms;
- 3 Push-buttons;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 173: Circuito.



Fonte: Imagem tirada de (Squids, 2017).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Sensor Ultrassônico com Buzzer e Led

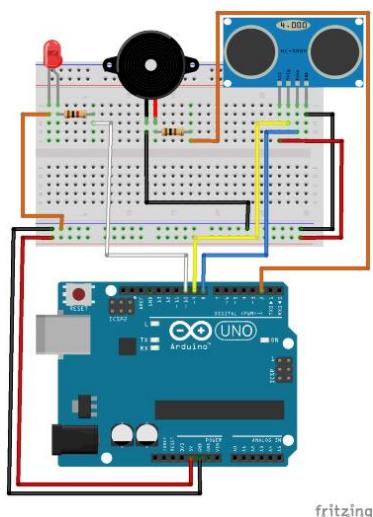
Neste experimento deve-se ligar um LED e um Buzzer quando houver aproximação de algum objeto ao sensor ultrassônico.

Materiais necessários:

- 1 Buzzer;
- 1 LED Vermelho;
- 1 Sensor Ultrassônico;
- 1 Resistor de 100 ohms e 1 Resistor de 330 ohms;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 174: Circuito.



Fonte: Imagem tirada de (Blog do Renato Lima, 2017).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Relógio com hora e data

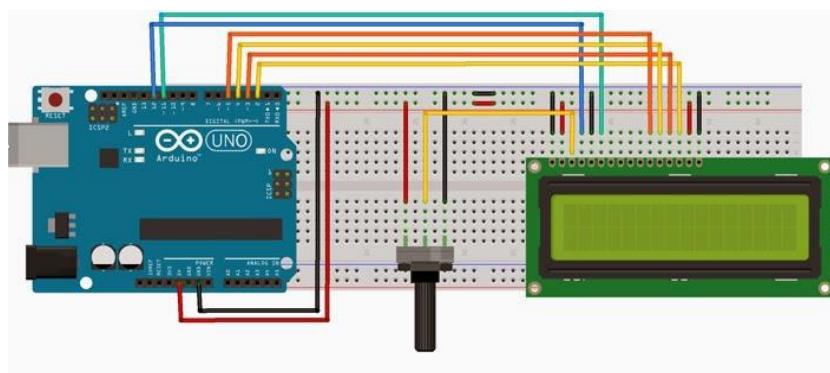
Este experimento consiste em fazer um relógio utilizando o LCD.

Materiais necessários:

- 1 Display LCD;
- 1 Potenciômetro;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 175: Circuito.



Fonte: Imagem tirada de (PIMENTA, 2014).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Sensor de distância com LEDs

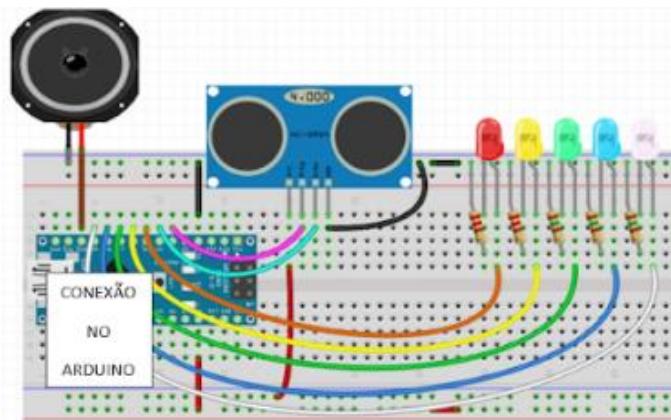
Neste experimento o sensor de distância ultrassônico irá identificar a distância de objetos. Defina uma distância para cada Led acender e soar um som diferente no buzzer a medida em que a distância mudar.

Materiais necessários:

- 5 LEDs;
- 1 Buzzer;
- 5 Resistores 330 ohms;
- 1 Sensor Ultrassônico;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 176: Circuito.



Fonte: Imagem tirada de (Eletrônica para artistas, 2017).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Sensor de distância e motor

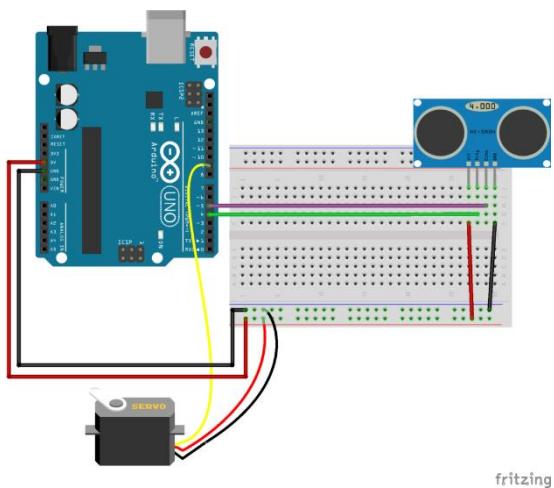
Neste projeto quando o sensor ultrassônico identificar algum objeto o servo motor deve se deslocar 45° e enviar para o monitor serial “aberto” e quando não identificar deve voltar a 0° e enviar para o monitor serial “fechado”.

Materiais necessários:

- 1 Servo Motor;
- 1 Sensor Ultrassônico;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Circuito:

Figura 177: Circuito.



Fonte: Imagem tirada de (FilipeFlop, 2019).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Sensor óptico reflexivo com botão de reset

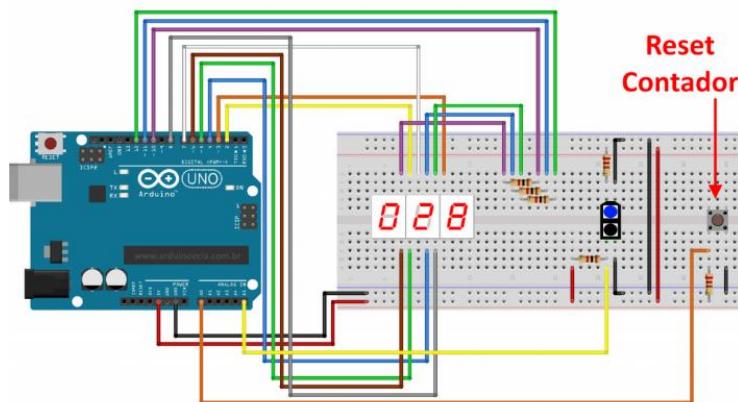
Neste experimento o contador é incrementado cada vez que o sensor óptico for acionado, ao ser acionado deve mostrar o resultado no display 7 segmentos, zerando o contador ao apertar o botão.

Materiais necessários:

- 3 Displays 7 segmentos;
- 1 Sensor óptico reflexivo;
- 1 Push-button;
- 3 resistores de 1K ohms, 1 resistor de 220 ohms e 2 resistores de 10K ohms;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 178: Círcuito.



Fonte: Imagem tirada de (Arduino e CIA, 2015).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Jogo da Tabuada

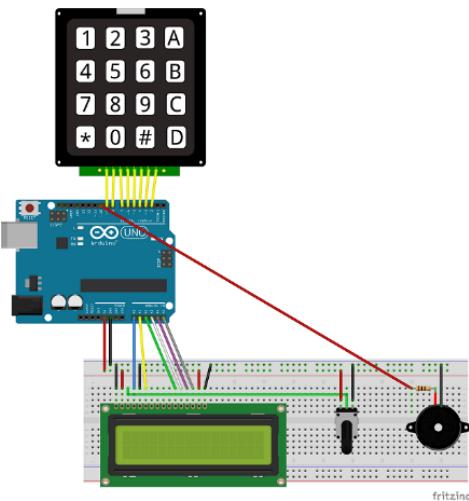
Neste experimento o sistema irá escolher dois números aleatórios para serem multiplicados, que serão exibidos no Display LCD, o usuário deve utilizar o Keypad para dar sua resposta onde o ENTER será o “#”, se a resposta estiver correta deve aparecer no Display LCD “Resposta Certa”, caso esteja errada o Buzzer deve ser ativado e deve aparecer no Display LCD “Resposta Incorreta”. Para retornar ao início é só apertar a tecla “#”.

Materiais necessários:

- 1 Keypad;
- 1 Buzzer;
- 1 Display LCD;
- 1 Potenciômetro;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Circuito:

Figura 179: Circuito.



Fonte: Imagem tirada de (Blog do José Cintra, 2016).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Servo motor com Sensor LDR

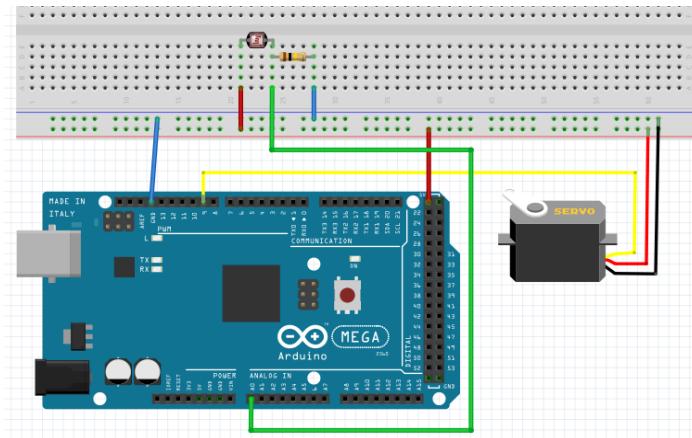
Nesse experimento o servo motor será controlado pelo sensor LDR, sua rotação dependerá da quantidade de luz recebida pelo sensor, determine quantos graus ele deve andar para cada quantidade de luz do LDR.

Materiais necessários:

- 1 Sensor LDR;
- 1 Servo motor;
- 1 Resistor de 10K ohms;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 180: Circuito.



Fonte: Imagem tirada de (Blog do DRK, 2016).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Movendo o Servo motor com botões

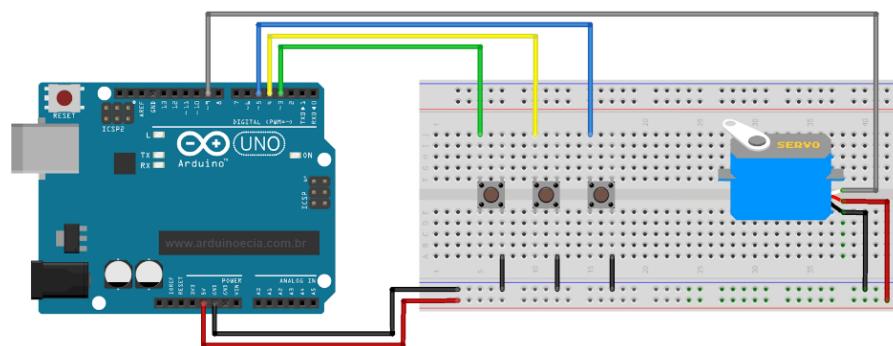
Nesse experimento ao pressionar o botão da esquerda, o servo se movimenta para “60 graus”, o botão central para “90 graus”, e o botão da direita para posição “120 graus”.

Materiais necessários:

- 1 Servo motor;
- 3 Push-buttons;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 181: Circuito.



Fonte: Imagem tirada de (Arduino e CIA, 2013).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Acelerando e desacelerando o motor com luz

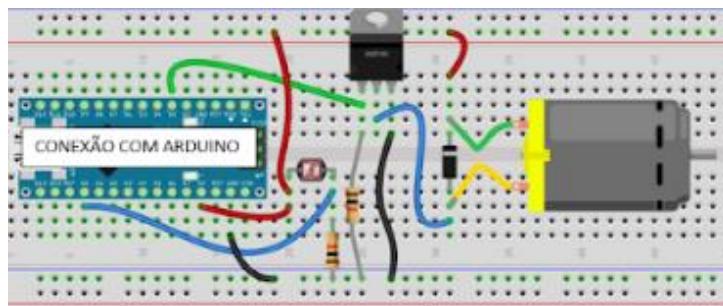
Neste experimento quanto mais luz o sensor LDR receber, maior será a velocidade do motor de passo e vice-versa.

Materiais necessários:

- 1 Motor de Passo;
 - 1 Diodo;
 - 2 Resistores de 10K ohms;
 - 1 Sensor LDR;
 - 1 Protoboard;
 - 1 Arduino Uno;
 - Jumpers diversos.

Circuito:

Figura 182: Circuito.



Fonte: Imagem tirada de (Eletrônica para artistas, 2017).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

LEDs e Sensor RGB

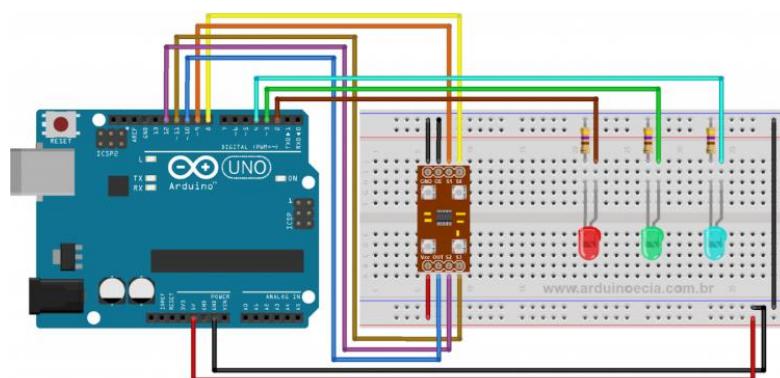
Inclua 3 LEDs nas cores correspondentes a RGB, para que acendam quando determinada cor for reconhecida pelo sensor e apresente o resultado também no monitor serial. Caso queira aumentar a quantidade de cores basta formar com os valores de RGB, adicionando LEDs correspondentes a elas.

Materiais necessários:

- 1 Sensor RGB;
- 1 Led Verde;
- 1 Led Vermelho;
- 1 Led Azul;
- 3 Resistores de 330 ohms;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 183: Circuito.



Fonte: Imagem tirada de (Arduino e CIA, 2014).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Fechadura eletrônica controlada por senha

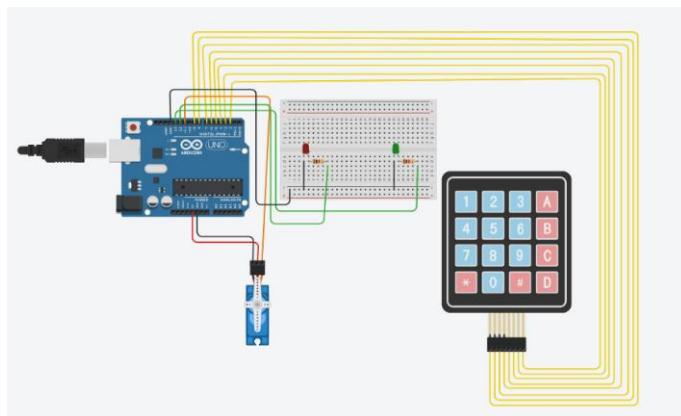
Nesse experimento quando o usuário digita uma senha no teclado, o Arduino processa uma informação verificando se a senha é válida ou não, caso seja válida o Led verde acende e o Servo motor gira, caso contrário o motor fica parado e acende o LED vermelho. Para retornar o motor na posição inicial basta clicar na tecla “*”.

Materiais necessários:

- 1 Keypad;
- 1 Led Verde;
- 1 Led Vermelho;
- 1 Servo Motor;
- 2 Resistores de 330 ohms;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Circuito:

Figura 184: Circuito.



Fonte: Imagem tirada de (Medium, 2019).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Keypad e Display 7 segmentos

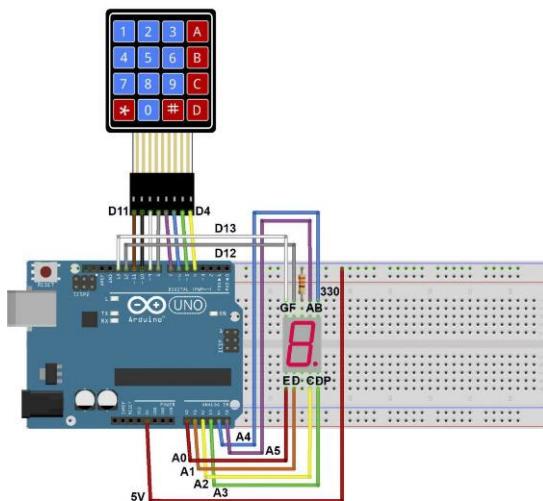
O objetivo desse experimento é identificar a tecla pressionada e mostrá-la no display 7 segmentos. Como no display não existe os caracteres * e #, identifique-os como outras letras, por exemplo, E para * e F para #.

Materiais necessários:

- 1 Keypad;
- 1 Display 7 segmentos;
- 1 Resistor de 330 ohms;
- 1 Protoboard;
- 1 Arduino Uno;
- Jumpers diversos.

Círcuito:

Figura 185: Circuito.



Fonte: Imagem tirada de (SHIGUEMORI, 2019).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Semáforo Interativo com Display 7 segmentos

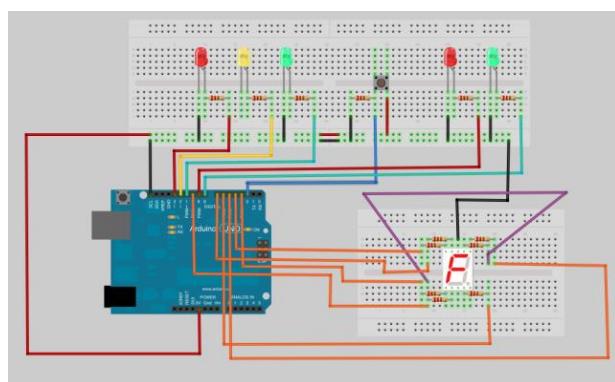
Assim como no experimento Semáforo Interativo, ao usuário apertar o botão imediatamente o sinal deve ficar amarelo e depois passar para vermelho o de automóveis e verde o de pedestre. O diferencial desse projeto é que quando o botão não for pressionado o sinal que está verde deve fechar em algum momento, assim antes de fechar utilize do display 7 segmentos para fazer essa contagem regressiva de 5 segundos, acendendo o LED amarelo e depois desse tempo acendendo o LED vermelho para carros e o verde para pedestres.

Materiais necessários:

- 2 LEDs verde;
- 2 LEDs vermelho;
- 1 LED amarelo;
- 1 Display 7 segmentos;
- 1 Push-button;
- 12 Resistores 220 ohms e 1 Resistor 10K ohms;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Circuito:

Figura 186: Circuito.



Fonte: Imagem tirada de (Laboratório de Garagem, 2012).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Sensor de Umidade e Display LCD

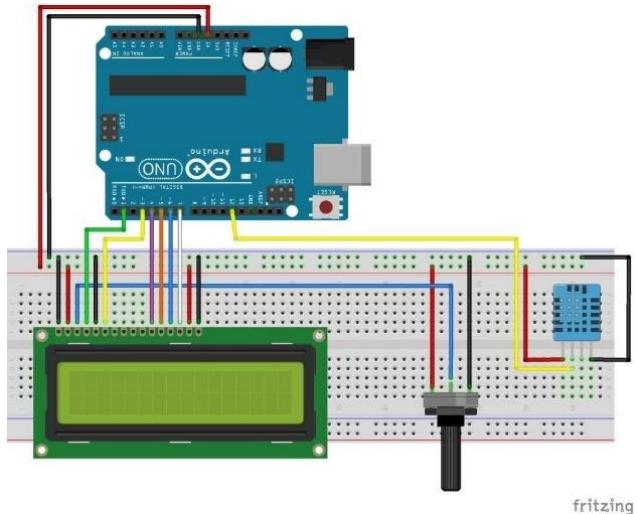
Neste experimento deve ser apresentado os resultados de umidade e temperatura lidas pelo sensor DHT22 no Display LCD.

Materiais necessários:

- 1 Display LCD;
 - 1 Potenciômetro;
 - 1 Sensor DHT11;
 - 1 Arduino Uno;
 - 1 Protoboard;
 - Jumpers diversos.

Circuito:

Figura 187: Circuito.



Fonte: Imagem tirada de (Baú da Eletrônica, 2017).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

DHT11 com Buzzer e Led

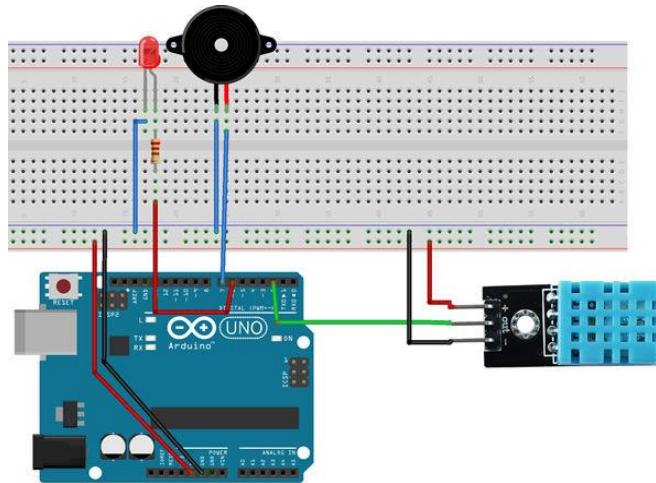
A leitura de temperatura e umidade devem ser feitas e armazenadas em variáveis que depois serão comparadas com a temperatura lida no momento do teste, caso a temperatura seja diferente o Led e o Buzzer devem ser ativados.

Materiais necessários:

- 1 Buzzer;
- 1 LED;
- 1 Sensor DHT11;
- 1 Resistor 220 ohms;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Círcuito:

Figura 188: Circuito.



Fonte: Imagem tirada de (Usinainfo, 2016).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

DHT22 com Sensor LDR e Sensor Ultrassônico

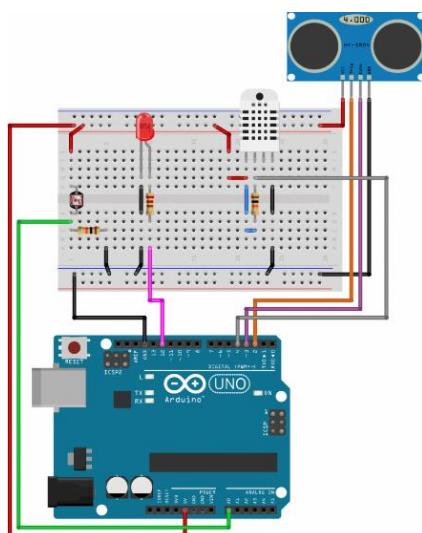
O objetivo desse experimento é ler os três sensores, assim se a temperatura lida pelo DHT22 estiver acima de 25 graus Celsius, a leitura do LDR indicar ambiente claro e a distância lida pelo sensor ultrassônico for menor que 30cm, o LED acende, caso contrário o Led apaga.

Materiais necessários:

- 1 Sensor Ultrassônico;
- 1 LED;
- 1 Sensor LDR;
- 1 Sensor DHT22;
- 1 Resistor 220 ohms e 2 Resistores de 10K ohms;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Circuito:

Figura 189: Circuito.



Fonte: Imagem tirada de (FilipeFlop, 2018).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Sensor DHT22 com alarme RGB

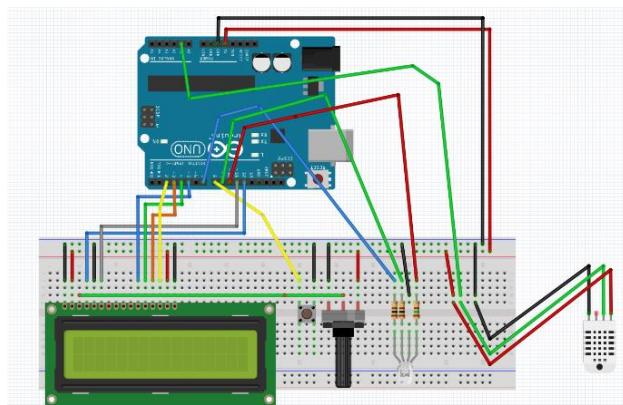
A função do sensor DHT22 é captar a temperatura e a umidade do ambiente (quando for apresentado no Display LCD deve-se ter um botão, para que quando for pressionado o visor que antes mostrava a temperatura, deve apresentar a umidade) e do Led RGB é ser um alarme onde a temperatura pode ser abaixo de 15° (LED deve ficar azul), entre 15° e 25° (LED deve ficar verde) e acima de 25° (LED deve ficar vermelho). Os valores de temperatura e umidade devem ser apresentados no display LCD e no monitor serial.

Materiais necessários:

- 1 Display LCD;
 - 1 Potenciômetro;
 - 1 Led RGB;
 - 1 Push-button;
 - 1 Sensor DHT22;
 - 3 Resistores de 220 ohms;
 - 1 Arduino Uno;
 - 1 Protoboard;
 - Jumpers diversos.

Circuito:

Figura 190: Circuito.



Fonte: Imagem tirada de (Squids, 2018).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Dado eletrônico

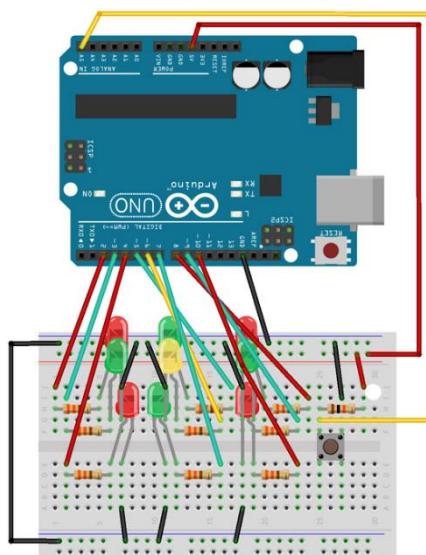
Um dado eletrônico é formado por 6 LEDs (cada um representa 1, então por exemplo liga-se dois LEDs, o valor do dado será 2) e um botão que ao pressioná-lo ele sorteia o valor apresentado nos LEDs.

Materiais necessários:

- 1 Push-button;
- 6 LEDs;
- 6 Resistores de 220 ohms e 1 Resistor 10K ohms;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Círcuito:

Figura 191: Círcuito.



Fonte: Imagem tirada de (Laboratório de Garagem, 2015).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Monitorando Luminosidade e Temperatura com LCD

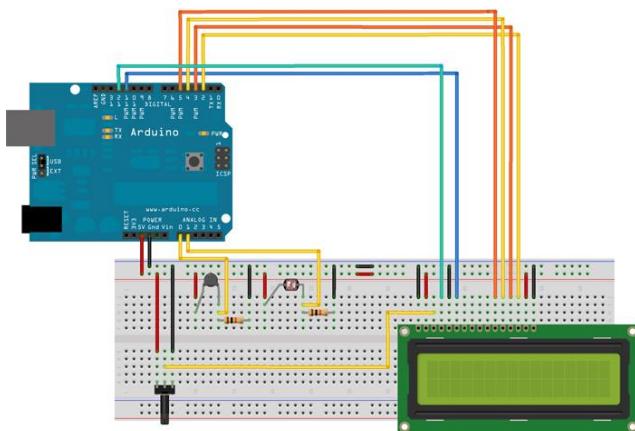
O objetivo deste experimento é ler a temperatura por meio do sensor NTC e a luminosidade pelo sensor LDR, que serão apresentados no Display LCD.

Materiais necessários:

- 1 Sensor LDR;
- 1 Sensor NTC;
- 1 Potenciômetro;
- 1 Display LCD;
- 2 Resistores de 10K ohms;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Círcuito:

Figura 192: Circuito.



Fonte: Imagem tirada de (Laboratório de Garagem, 2016).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

Jogo da memória - Genius

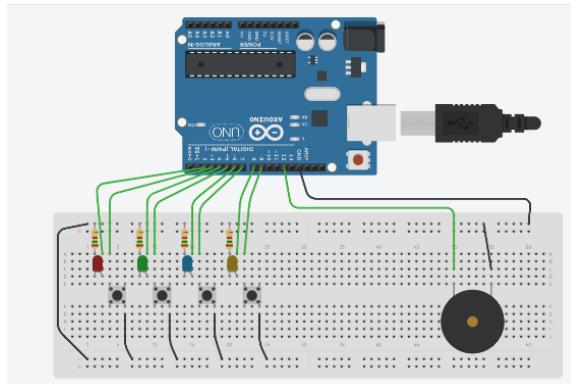
Neste experimento será feito um jogo, formado por 4 botões e 4 LEDs que acendem em uma sequência aleatória, com sons definidos para cada um dos LEDs. O jogador deverá clicar nos botões de acordo com a sequência gerada pela Arduino. A cada jogada, a sequência é acrescida de um novo item, ou seja, começa acendendo apenas um Led, caso a pessoa acerte deve acender o anterior e o novo, assim totalizando em 2 passos e assim por diante. Quando o usuário errar o passo, o jogo deve ser reinicializado.

Materiais necessários:

- 1 Led Azul;
- 1 Led Vermelho;
- 1 Led Verde;
- 1 Led Amarelo;
- 4 Push-buttons;
- 1 Buzzer;
- 4 Resistores de 220 ohms;
- 1 Arduino Uno;
- 1 Protoboard;
- Jumpers diversos.

Círcuito:

Figura 193: Circuito.



Fonte: Imagem tirada de (Squids, 2020).

Anote sobre o experimento (resultados, dúvidas, curiosidades etc.).

PROJETOS

Cubo de Led

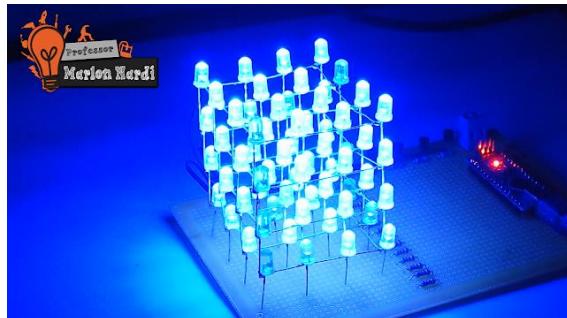
O cubo de Led é composto por uma matriz tridimensional de LEDs formando um cubo, cuja função é mostrar animações em 3D. Neste projeto ele pode atingir uma variedade de tamanhos.

Regras do Projeto:

- Ter mínimo de 3x3x3;
- Apresentar mínimo de 3 animações, como por exemplo: acender todos os LEDs, acender um Led de cada vez, formar letras, acender linha por linha, entre outros;
- Mudar a animação cada vez que o botão for pressionado.

Exemplos de Projetos:

Figura 194: Exemplo Professor Marlon Nardi.



Fonte: Imagem tirada de (Professor Marlon Nardi, 2016).

Anote sobre o projeto (resultados, dúvidas, curiosidades etc.).

Controle de Acesso usando Leitor RFID

RFID é um sistema de identificação por meio de radiofrequência que permite acesso a diversos ambientes. Ele utiliza ondas de rádio para a identificação das tags que emite um código exclusivo para cada acessório que for ser identificado. Neste projeto por meio leitor RFID sempre que for identificado a tag cadastrada a trava será liberada, impedindo a passagem daquelas não cadastradas. O Arduino emparelhará um cartão ou um chaveiro para destravar.

Regras do Projeto:

- Apresentar os resultados via Monitor Serial, Display LCD e por uma trava (ou girando um servo motor);
- Deve ter uma contagem (mostrar no Display LCD) para bloquear a trava novamente logo após ela ser liberada, caso ela não seja liberada deve haver a contagem mesmo assim, para que o usuário tenha uma segunda tentativa de desbloqueá-la;
- Caso após 4 tentativas ela não for destravada, o buzzer deve ser ativado como forma de alarme;
- Ter no mínimo dois objetos para serem lidos pelo RFID, um fará com que a trava libere e outro não será aceito (não estará cadastrado).

Exemplos de Projetos:

Figura 195: Exemplo Usaininfo Eletrônica e Robótica.



Fonte: Imagem tirada de (YouTube, 2017).

Anote sobre o projeto (resultados, dúvidas, curiosidades etc.).

Árvore de Natal

Para modernizar o natal, monte sua própria árvore, tendo como base uma placa fenolite que será a árvore onde serão pendurados os pisca-piscas (LEDs), o tamanho fica a critério de quem monta.

Regras do Projeto:

- Ter no mínimo 12 LEDs na “árvore”;
- Cortar a placa fenolite no formato de uma árvore e no tamanho ideal para que caiba todos os LEDs;
- Para acompanhar a árvore de natal, faça com que o buzzer toque no mínimo duas músicas de natal, sendo alteradas por um botão.

Exemplos de Projetos:

Figura 196: Exemplo FilipeFlop,



Fonte: Imagem tirada de (FilipeFlop, 2015).

Anote sobre o projeto (resultados, dúvidas, curiosidades etc.).

Alarme inteligente

Este sistema terá como seu principal componente o sensor de movimento PIR, que detecta qualquer movimento no ambiente, seu funcionamento é a partir de um sensor eletrônico que mede a luz infravermelha irradiada de objetos. O objetivo deste projeto é controlar este sensor e disparar diversas ações ao detectar movimentos.

Regras do Projeto:

- Ao reconhecer o movimento, no mínimo 5 LEDs devem acender (com 2 formas de piscar diferentes, controladas por um botão, que será definido pela pessoa que faz o uso do alarme) como forma de sinal;
- Um buzzer deve ser ativado ao detectar movimento, que só será desativado por um botão (este botão deve desativar o buzzer e apagar os LEDs).

Exemplos de Projetos:

Figura 197: Exemplo como fazer as coisas.



Fonte: Imagem tirada de (Como fazer as coisas, 2013).

Anote sobre o projeto (resultados, dúvidas, curiosidades etc.).

Sistema de Irrigação

Este Projeto tem como objetivo cuidar de uma planta de forma autônoma, eliminando qualquer risco de falta ou excesso de água e regando-a quando necessário. Dois componentes fundamentais são o Módulo Sensor de Umidade do Solo e a Mini Bomba de Água RS-385.

Regras do Projeto:

- Adicione um Display LCD, que deve avisar com mensagens quando a planta estiver sendo irrigada e quando ela já foi irrigada;
- Pesquise a porcentagem de água que a planta utilizada no experimento necessita, proporcionando a ela saúde e beleza (utilize o sensor de umidade do solo para controlar esta porcentagem).

Exemplos de Projetos:

Figura 198: Exemplo Matheus Gebert Straub.



Fonte: Imagem tirada de (UsinalInfo, 2019).

Anote sobre o projeto (resultados, dúvidas, curiosidades etc.).

Dispenser automático de álcool em gel

Durante a pandemia, momento fundamental para se prevenir ao máximo, deve-se seguir uma série de cuidados, inclusive higienizar as mãos com álcool em gel. Pensando nisso, esse projeto consiste em integrar o sensor ultrassônico ao servo motor, este irá pressionar a tampa do álcool despejando-o na mão do usuário.

Regras do Projeto:

- O servo será ativado quando a pessoa aproximar a mão ao sensor ultrassônico (deixe uma distância pequena para ser detectado);
- Para que o álcool seja despejado o servo deve girar, mas logo voltar na posição inicial, evitando desperdícios;
- Crie um suporte para que tudo esteja conectado.

Exemplos de Projetos:

Figura 199: Exemplo Luiz Fernando.



Fonte: Imagem tirada de (FilipeFlop, 2020).

Anote sobre o projeto (resultados, dúvidas, curiosidades etc.).

Robô seguidor de linha e desvia de obstáculos

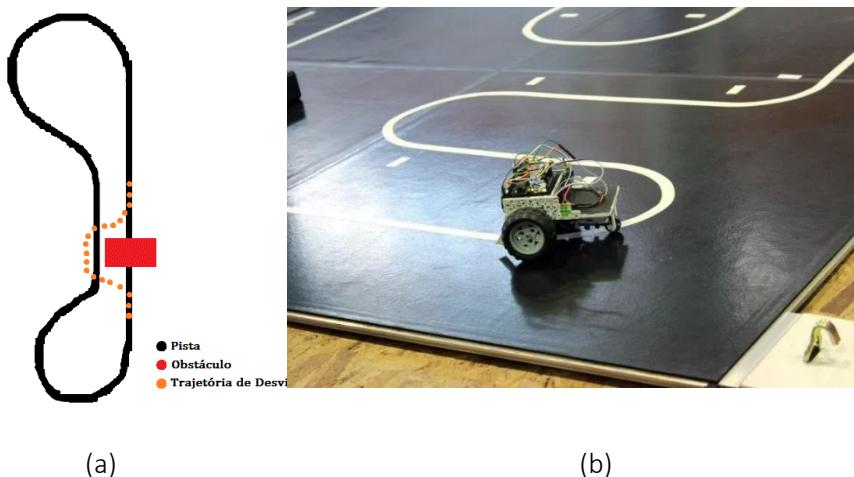
Robôs seguidores de linha e que desviam de obstáculos são pré-programados, sua função é detectar onde existe um caminho, no caso do projeto uma linha branca ou preta, através da leitura de sensores infravermelho e desviar de obstáculos que estiverem no meio desse caminho das linhas identificado pelo sensor ultrassônico.

Regras do Projeto:

- Monte um trajeto para o seu robô, pode ser em uma madeira ou mesmo no chão da sala de aula marcado com fita (que irá variar ser branca ou preta dependendo do fundo que tiver);
- Deve-se ter um obstáculo no mínimo no trajeto, ele deve desviar do objeto e retornar para o seu caminho;
- Além de um obstáculo, deve seguir como base de trajeto a imagem 200 (a) do exemplo.

Exemplos de Projetos:

Figura 200: (a) Trajeto e (b) Exemplo Gradimildo Candido.



Fonte: (a) Imagem tirada de (Laboratório de Garagem, 2014) e (b) Imagem tirada de (Vida de Silício, 2018).

Anote sobre o projeto (resultados, dúvidas, curiosidades etc.).

Controlando Projeto com Controle Remoto

O objetivo desse projeto é controlar um circuito com controle remoto (Kit Infravermelho + Controle Remoto) que funciona com tecnologia infravermelha. Para o bom funcionamento as teclas do controle devem ser clonadas, para em seguida definir ações de acordo com a tecla pressionada.

Regras do Projeto:

- O controle deve comandar no mínimo 3 LEDs, um Buzzer e um elemento surpresa;
- Para esse projeto deve-se ter um elemento surpresa (Ex: conectar um Led RGB ao controle, que a cada botão que for apertado, uma cor será acionada);
- Os LEDs e o buzzer que devem estar conectados ao controle, precisam ser ativados e desativados pelo mesmo botão.

Exemplos de Projetos:

Figura 201: Exemplo Euler Oliveira.



Fonte: Imagem tirada de (Master Walker, 2019).

Anote sobre o projeto (resultados, dúvidas, curiosidades etc.).

Acender lâmpada com palmas

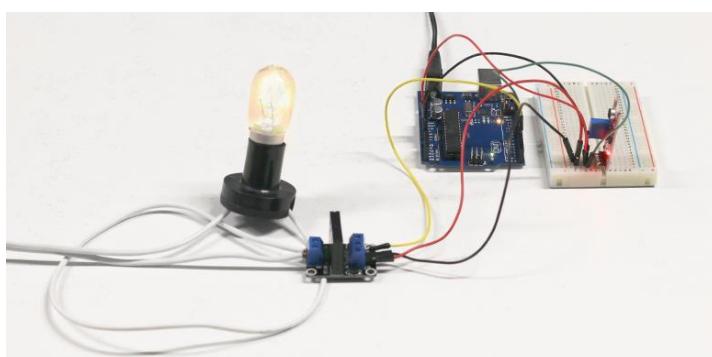
Nesse projeto vamos desenvolver um sistema para acender uma lâmpada com palmas, utilizando um Módulo Relé 5v de 1 canal e um Módulo Sensor de Som que será responsável por captar o barulho da palma e ativar a luz.

Regras do Projeto:

- A lâmpada deve acender com uma palma e apagar com duas;
- Monte uma estrutura para posicionar a lâmpada;
- Estude o Módulo Sensor de Som antes de iniciar o projeto, faça testes com ele separado integrando-o depois.

Exemplos de Projetos:

Figura 202: Exemplo André Rocha.



Fonte: Imagem tirada de (FilipeFlop, 2018).

Anote sobre o projeto (resultados, dúvidas, curiosidades etc.).

Separando objetos por cor com braço mecânico

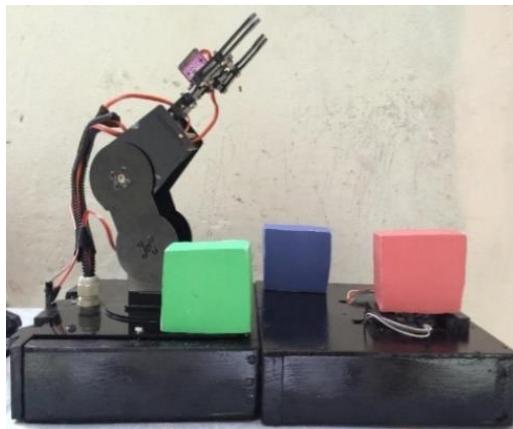
O principal objetivo deste projeto é automatizar um braço mecânico de modo que ele diferencie a cor das peças, através de um sensor de cor RGB, colocando cada peça na caixa de suas respectivas cores.

Regras do Projeto:

- O braço robótico deve identificar no mínimo as principais cores do sensor RGB, vermelho, verde e azul;
- Deve-se ter 3 caixas, cada uma da cor do objeto, e 6 objetos (2 objetos para cada cor);
- No final deve existir 2 objetos dentro de cada caixa.

Exemplos de Projetos:

Figura 203: Exemplo Lucas Augusto Vieira Brito.



Fonte: Imagem tirada de (Doc Player, 2016).

Anote sobre o projeto (resultados, dúvidas, curiosidades etc.).

Painel de Senhas de Atendimento

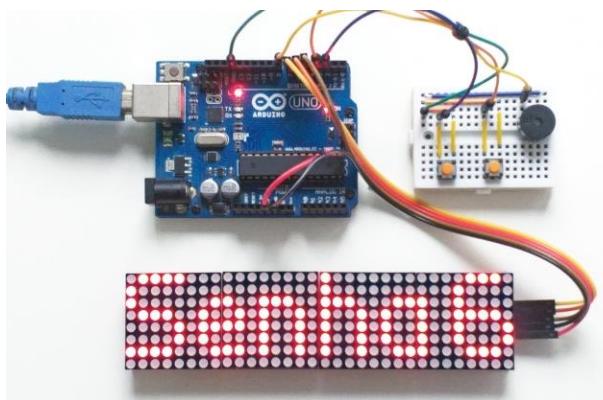
O Painel de Senhas é normalmente utilizado para organizar o atendimento ao público em hospitais, comércios, entre outros. Neste projeto será utilizado uma matriz de LED, onde será apresentado o número de senha das pessoas, este será incrementado ou decrementado por meio de 2 botões (cada um para uma função).

Regras do Projeto:

- O projeto será composto por 3 botões, um para incrementar o número, outro para decrementar e outro para reiniciar o painel de LED;
- Será permanente a mensagem “Senha” o que deve ser alterado é apenas o valor. Ex: “Senha 1” após clicar o botão será “Senha 2”;
- Todas as vezes que o número que será apresentado no painel de LED for modificado, o buzzer deve ser ativado por um curto tempo e logo ser desativado, como forma de aviso da troca.

Exemplos de Projetos:

Figura 204: Exemplo Arduino e CIA.



Fonte: Imagem tirada de (Arduino e CIA, 2019).

Anote sobre o projeto (resultados, dúvidas, curiosidades etc.).

BIBLIOGRAFIA

A TRIBUTE TO JOSEPH ENGELBERGER. **Unimate.** Disponível em: <https://www.robots.org/joseph-engelberger/unimate.cfm>. Acesso em: 9 abr. 2020.

AECX ROBOT. **Aula 6 - O Multímetro.** Disponível em: <http://aecxrobot.blogspot.com/p/aula-6-o-multi.html>. Acesso em: 20 jun. 2020.

ÁGUA RARA. **Clepsidra: como fazer um relógio d'água.** Disponível em: <http://aguararabr.blogspot.com/2015/08/clepsidra-como-fazer-um-relogio-dagua.html>. Acesso em: 2 jun. 2020.

ALIBABA. **W-ROBOT linear manipulador robô cartesiano XYZ braço VFA01.** Disponível em: <https://portuguese.alibaba.com/product-detail/w-robot-linear-manipulator-xyz-cartesian-robot-arm-vfa01-1873123198.html>. Acesso em: 9 jun. 2020.

ARDUINO BY MYSELF. **Sistema de Irrigação.** Disponível em: <https://arduinobymyself.blogspot.com/2013/09/sistema-de-irrigacao.html>. Acesso em: 5 ago. 2020.

ARDUINO E CIA. **Como usar o display 7 segmentos com Arduino.** Disponível em: <https://www.arduinoecia.com.br/display-7-segmentos-com-arduino/>. Acesso em: 10 jul. 2020.

ARDUINO E CIA. **Como usar um servo motor com Arduino.** Disponível em: <https://www.arduinoecia.com.br/controlando-um-servo-motor-com-arduino/>. Acesso em: 16 jul. 2020.

ARDUINO E CIA. **Contador com botão de reset.** Disponível em: <https://www.arduinoecia.com.br/contador-lcd-display-7-segmentos-reset/>. Acesso em: 16 jul. 2020.

ARDUINO E CIA. **Painel de senhas com Arduino.** Disponível em: <https://www.arduinoecia.com.br/painel-de-senhas-com-arduino-max7219/>. Acesso em: 11 ago. 2020.

ARDUINO E CIA. **Sensor de cor TCS230 / TCS3200 com Arduino.** Disponível em: <https://www.arduinoecia.com.br/sensor-de-reconhecimento-de-cor-tcs230-tcs3200/>. Acesso em: 17 jul. 2020.

ARDUINO E CIA. **Medidor de distância com sensor ultrassônico HC-SR04.** Disponível em: <https://www.arduinoecia.com.br/medidor-de-distancia-sensor-ultrassonico-hc-sr04-arduino/>. Acesso em: 29 jun. 2020.

ARDUINO E CIA. **Sensor óptico reflexivo TCRT5000 com Arduino.** Disponível em: <https://www.arduinoecia.com.br/sensor-optico-reflexivo-tCRT5000-arduino/>. Acesso em: 29 jun. 2020.

ARDUINO PORTUGAL. O que é o Arduino UNO? Disponível em:
<https://www.arduinoportugal.pt/o-que-e-o-arduino/>. Acesso em: 27 abr. 2020.

ARDUINO. Display de 7 Segmentos. Disponível em:
<https://ardudino.wordpress.com/2016/03/06/display-de-7-segmentos/>. Acesso em: 10 jul. 2020.

ARDUINO. Documentação de Referência da Linguagem Arduino. Disponível em:
<https://www.arduino.cc/reference/pt/>. Acesso em: 1 jun. 2020.

ARDUINO. switch...case. Disponível em:
<https://www.arduino.cc/reference/pt/language/structure/control-structure/switchcase/>. Acesso em: 26 mai. 2020.

ATHOS ELECTRONICS. Código de Cores de Resistores. Disponível em:
<https://athoselectronics.com/codigo-de-cores-de-resistores/>. Acesso em: 18 jun. 2020.

ATHOS ELECTRONICS. LDR – O que é e como efetuar leituras. Disponível em:
<https://athoselectronics.com/ldr-o-que-e-como-funciona/>. Acesso em: 26 jun. 2020.

ATHOS ELECTRONICS. O que é LED? Para que serve? Disponível em:
<https://athoselectronics.com/o-que-e-led-diodo-emissor-luz/>. Acesso em: 24 jun. 2020.

ATHOS ELETRONICS. Código de Cores de Resistores. Disponível em:
<https://athoselectronics.com/codigo-de-cores-de-resistores/>. Acesso em: 6 mai. 2020.

AUTOCORE. Controlando o Arduino com Controle Remoto. Disponível em:
<https://autocorerobotica.blog.br/controlando-o-arduino-com-controle-remoto/>. Acesso em: 6 ago. 2020.

AUTODESK TINKERCAD. Da mente ao projeto em minutos. Disponível em:
<https://www.tinkercad.com/>. Acesso em: 8 mai. 2020.

BAÚ DA ELETRÔNICA. Como Utilizar o Sensor de Cor RGB TCS34725. Disponível em:
<http://blog.baudaelectronica.com.br/utilizando-o-sensor-de-cor-tcs34725/>. Acesso em: 8 jul. 2020.

BAÚ DA ELETRÔNICA. Conhecendo o LED RGB. Disponível em:
<http://blog.baudaelectronica.com.br/leds-rgb/>. Acesso em: 6 jul. 2020.

BAÚ DA ELETRÔNICA. Controlando Motor de Passo com Arduino + Driver ULN2003. Disponível em:
<http://blog.baudaelectronica.com.br/motor-de-passo-com-arduino/>. Acesso em: 1 jul. 2020.

BAÚ DA ELETRÔNICA. DHT11: Sensor de Umidade e Temperatura com Arduino Uno. Disponível em:
<http://blog.baudaelectronica.com.br/dht11-com-arduino/>. Acesso em: 28 jul. 2020.

BAÚ DA ELETRÔNICA. Utilizando o Teclado Matricial 4x4 no Arduino Uno. Disponível em:
<http://blog.baudaelectronica.com.br/teclado-matricial-4x4-com-arduino/>. Acesso em: 9 jul. 2020.

BAÚ DA ELETRÔNICA. Controle de Luminosidade de LED com Potenciômetro. Disponível em:
<http://blog.baudaelectronica.com.br/controle-de-luminosidade-de-led/>. Acesso em: 19 jun. 2020.

BLOG DO JOSÉ CINTRA. **Arduino – Jogo da Tabuada com LCD, Teclado e Buzzer.** Disponível em: <http://josecintra.com/blog/arduino-lcd-teclado-buzzer/>. Acesso em: 16 jul. 2020.

BLOG DO PROFESSOR CARLÃO. **Iniciação à Robótica Aula 4.** Disponível em: <https://www.blogdoprofessorcarlao.com.br/2009/12/iniciacao-robotica-aula-4.html>. Acesso em: 10 jun. 2020.

BLOG DO RENATO DE LIMA. **Sensor de Distância com Arduino.** Disponível em: <http://renatodelima.com.br/2017/04/sensor-de-distancia-com-arduino/>. Acesso em: 15 jul. 2020.

BLOG.UCL. **O que é robótica e por que você precisa se importar?** Disponível em: <https://blog.ucl.br/o-que-e-robotica-e-por-que-voce-precisa-se-importar/>. Acesso em: 2 abr. 2020.

BLOGSPOT. **Controle e Automação Industrial III.** Disponível em: <http://controleeautomacaoindustrial3.blogspot.com/2014/05/aula-123-robo-do-tipo-cartesiano.html>. Acesso em: 9 jun. 2020.

BÓSON TREINAMENTO EM TECNOLOGIA. **Programação para Arduino – Condicional if else.** Disponível em: <http://www.bosontreinamentos.com.br/elettronica/arduino/programacao-para-arduino-condicional-if-else/>. Acesso em: 27 mai. 2020.

BÓSON TREINAMENTOS EM TECNOLOGIA. **Como funciona um Potenciômetro.** Disponível em: <http://www.bosontreinamentos.com.br/elettronica/cursode-elettronica/como-funciona-um-potenciometro/>. Acesso em: 20 jun. 2020.

BRUM, M. N. **Introdução à Robótica Educativa.** [s.i]. 2011.

BUSINESS INSIDER. **Waymo could be worth as much \$175 billion — here's a brief history of the Google Car project.** Disponível em: <https://www.businessinsider.com/google-car-project-history-2018-8>. Acesso em: 4 jun. 2020.

CAR BODY DESIGN. **Toyota i Real Concept.** Disponível em: <https://www.carbodydesign.com/gallery/2007/10/20-toyota-i-real-concept/3/>. Acesso em: 15 jun. 2020.

CITISYSTEMS. **Os 6 Principais Tipos de Robôs Industriais.** Disponível em: <https://www.citisystems.com.br/tipos-de-robos/>. Acesso em: 14 abr. 2020.

COMO FAZER AS COISAS. **Como fazer um Alarme com Arduino e sensor de movimentos PIR.** Disponível em: <http://www.comofazerascoisas.com.br/como-fazer-um-alarme-com-arduino-sensor-de-movimentos-pir.html>. Acesso em: 5 ago. 2020.

COMO FAZER AS COISAS. **Projeto Arduino com sensor de luz LDR com saída de leitura em leds.** Disponível em: <http://www.comofazerascoisas.com.br/projeto-arduino-sensor-de-luz-ldr-com-leds.html>. Acesso em: 14 jul. 2020.

COMO FAZER AS COISAS. **Potenciômetro, o que é, para que serve, tipos, aplicações e como funciona.** Disponível em: <http://www.comofazerascoisas.com.br/potenciometro-o-que-e-para-que-servi-e-como-funciona.html>. Acesso em: 25 jun. 2020.

COMO FAZER AS COISAS. **Projeto Arduino, como emitir sons com o buzzer.** Disponível em: <http://www.comofazerascoisas.com.br/projeto-arduino-como-emitir-sons-com-o-buzzer.html>. Acesso em: 26 jun. 2020.

DIWO. Variáveis em Arduino. Disponível em: <http://diwo.bq.com/pt-pt/variaveis-em-arduino/>. Acesso em: 1 jun. 2020.

DOCPLAYER. BRAÇO MECÂNICO AUTOMATIZADO USANDO UM CONTROLADOR ARDUINO GUIADO POR UM SENSOR DE COR RGB. Disponível em: <https://docplayer.com.br/11126971-Titulo-braco-mecanico-automatizado-usando-um-controlador-arduino-guiado-por-um-sensor-de-cor-rgb.html>. Acesso em: 11 ago. 2020.

DOCPLAYER. Robótica Robôs não industriais. Disponível em: <https://docplayer.com.br/5190772-Robotica-4-robos-nao-industriais.html>. Acesso em: 5 jun. 2020.

DRK BLOG. PRUEBA DE MOTOR SERVO CON ARDUINO. Disponível em: <http://blog.drk.com.ar/2016/prueba-de-motor-servo-con-arduino>. Acesso em: 16 jul. 2020.

EARTHSENSE. Revolutionizing Agriculture. Disponível em: <https://www.earthsense.co/>. Acesso em: 16 jun. 2020.

EDUCAÇÃO FINAL. História da robótica. Disponível em: http://www.citi.pt/educacao_final/trab_final_inteligencia_artificial/historia_da_robotica.html. Acesso em: 7 abr. 2020. 7

ELETROGATE. Sensores DHT11 e DHT22: Guia básico dos sensores de umidade e temperatura. Disponível em: <https://blog.eletrogate.com/sensores-dht11-dht22/>. Acesso em: 13 jul. 2020.

ELETROGATE. Servo motor com arduino: Conheça aplicações e aprenda a usar. Disponível em: <https://blog.eletrogate.com/servo-motor-para-aplicacoes-com-arduino/>. Acesso em: 30 jun. 2020.

ELETRÔNICA PARA ARTISTAS. Arduino 11: Música com bibliotecas. Disponível em: <http://eletronicaparaartistas.com.br/category/uncategorized/page/2/>. Acesso em: 17 jul. 2020.

ELETRÔNICA PARA ARTISTAS. Experimento 36 – Arduino com sensor de distância. Disponível em: <http://eletronicaparaartistas.com.br/experimento-36-arduino-com-sensor-de-distancia/>. Acesso em: 16 jul. 2020.

ELETRÔNICA PARA TODOS. Entendendo e controlando motores de passo com driver ULN2003 e Arduino. Disponível em: <https://eletronicaparatodos.com/entendendo-e-controlando-motores-de-passo-com-driver-uln2003-e-arduino-rodruino-board/>. Acesso em: 1 jul. 2020.

ELETRÔNICA PARA TODOS. Simulando um "semáforo de trânsito de carros e pedestres" com arduino. Disponível em: <https://eletronicaparatodos.com/simulando-um-semaforo-de-transito-de-carros-e-pedestres-com-arduino/>. Acesso em: 14 jul. 2020.

EMBARCADOS. Arduino UNO. Disponível em: <https://www.embarcados.com.br/arduino-uno/>. Acesso em: 25 abr. 2020.

EMBARCADOS. Usando as saídas PWM do Arduino. Disponível em:

<https://www.embarcados.com.br/pwm-do-arduino/>. Acesso em: 17 jun. 2020.

EPSON. Robô Epson Scara T3. Disponível em: <https://epson.com.br/Para-empresas/Rob%C3%B3tica/Robs-SCARA/Rob%C3%84s-SCARA/Rob%C3%84-Epson-Scara-T3/p/RT3-401SS-B>. Acesso em: 8 jun. 2020.

FAZEDORES. 5 Projetos Legais com Arduino e LEDs. Disponível em: <https://blog.fazedores.com/top-5-projetos-com-arduino-e-leds/>. Acesso em: 31 jul. 2020.

FERNÁNDEZ-PACHECO, A. S. V; TOME, Antonio García; Robótica Educativa. 1. ed. [S.I.]: RA-MA S.A. Editorial y Publicaciones, 2015. p. 32-43. UFSJ. **Aplicações da Robótica.** Disponível em: https://ufsj.edu.br/orcv/aplicacoes_da_robotica.php. Acesso em: 13 abr. 2020.

FILIPE FLOP. Controlando um Motor de Passo 5v com Arduino. Disponível em: <https://www.filipeflop.com/blog/controlando-um-motor-de-passo-5v-com-arduino/>. Acesso em: 1 jul. 2020.

FILIPE FLOP. Controle de Acesso usando Leitor RFID com Arduino. Disponível em: <https://www.filipeflop.com/blog/controle-acesso-leitor-rfid-arduino/>. Acesso em: 4 ago. 2020.

FILIPE FLOP. Dispenser automático de álcool em gel. Disponível em: <https://www.filipeflop.com/blog/dispenser-automatico-de-alcool-em-gel/>. Acesso em: 5 ago. 2020.

FILIPE FLOP. Identifique cores com o Sensor de Cor TCS3200 e Arduino. Disponível em: <https://www.filipeflop.com/blog/sensor-de-cor-tcs3200-rgb-arduino/>. Acesso em: 8 jul. 2020.

FILIPE FLOP. Monte sua Árvore de Natal com Arduino. Disponível em: <https://www.filipeflop.com/blog/arvore-de-natal-com-arduino/>. Acesso em: 4 ago. 2020.

FILIPE FLOP. Troque a cor das luzes. Disponível em: <https://www.filipeflop.com/universidade/kit-maker-arduino/projeto-7-troque-a-cor-das-luzes/>. Acesso em: 6 jul. 2020.

FILIPE FLOP. Usando um sensor de som para acender a luz batendo palmas. Disponível em: <https://www.filipeflop.com/blog/sensor-de-som-acender-luz-palmas/>. Acesso em: 6 ago. 2020.

FILIPE FLOP. Workshop Arduino para iniciantes – Arduino Day 2018. Disponível em: <https://www.filipeflop.com/blog/workshop-arduino-para-iniciantes/>. Acesso em: 28 jul. 2020.

FILIPE FLOP. Como conectar o Sensor Ultrassônico HC-SR04 ao Arduino. Disponível em: <https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>. Acesso em: 29 jun. 2020.

FILIPE FLOP. Motor DC 3V. Disponível em: <https://www.filipeflop.com/produto/mini-motor-dc-3v/>. Acesso em: 19 jun. 2020.

FLIPSNACK. Criador de Capa de Livro Online Grátis. Disponível em: <https://www.flipsnack.com/bp/book-cover-maker>. Acesso em: 1 abr. 2020

GUIA ESTUDO. Resistores. Disponível em: <https://www.guiastudo.com.br/resistores>. Acesso em: 18 jun. 2020.

IDEA. 10 projetos que você pode fazer com Arduino. Disponível em: <https://www.ideapd.com.br/10-projetos-arduino/>. Acesso em: 4 ago. 2020.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO - CÂMPUS JACAREÍ. Resumo Expandido CONICT. Jacareí, 2019.

INSTRUCTABLES CIRCUITS. **Learn to Design With Code Using Tinkercad Codeblocks: a Quick-Start Guide.** Disponível em: <https://www.instructables.com/id/Learn-to-Design-With-Code-Using-Tinkercad-Codeblock/>. Acesso em: 8 mai. 2020.

LABORATÓRIO DE ELETRÔNICA. **Robôs de Classe.** Disponível em: <http://labdeeletronica.com.br/robos-de-classe/>. Acesso em: 5 jun. 2020.

LABORATÓRIO DE GARAGEM. **Alarme de Temperatura com Buzzer e leds indicadores.** Disponível em: <http://labdegaragem.com/profiles/blogs/alarme-de-temperatura-com-buzzer-e-leds-indicadores>. Acesso em: 15 jul. 2020.

LABORATÓRIO DE GARAGEM. **Dado eletrônico.** Disponível em: http://arduino.labdegaragem.com/Guia_preview/EK_15_dado.html. Acesso em: 30 jul. 2020.

LABORATÓRIO DE GARAGEM. **Monitorando Luminosidade e Temperatura com LCD.** Disponível em: http://arduino.labdegaragem.com/Guia_preview/SMK_6_luminosidade_temperatura.html. Acesso em: 30 jul. 2020.

LABORATÓRIO DE GARAGEM. **Projetos.** Disponível em: <http://labdegaragem.com/page/projetos-1>. Acesso em: 31 jul. 2020.

LABORATÓRIO DE GARAGEM. **Semáforo com Arduino e display 7 segmentos.** Disponível em: <http://labdegaragem.com/profiles/blogs/sem-faro-com-arduino-e-display-7-segmentos>. Acesso em: 27 jul. 2020.

LABORATÓRIO DE GARAGEM. **Sistema de Irrigação Simples.** Disponível em: <http://labdegaragem.com/profiles/blogs/sistema-de-irriga-o-simples>. Acesso em: 5 ago. 2020.

LABORATÓRIO DE GARAGEM. **Tutorial Arduino - Motor de Passo 28BYJ-48 + driver ULN2003.** Disponível em: <http://labdegaragem.com/forum/topics/tutorial-arduino-motor-de-passo-28byj-48-driver-uln2003?id=6223006%3ATopic%3A521405&page=2>. Acesso em: 1 jul. 2020.

LABORATÓRIO DE GARAGEM. **Tutorial: Carrinho seguidor de linha que desvia de obstáculos com plataforma Zumo e Arduino.** Disponível em: <http://labdegaragem.com/profiles/blogs/tutorial-carrinho-seguidor-de-linha-que-desvia-de-obstaculos-com->. Acesso em: 10 ago. 2020.

LABORATÓRIO DE GARAGEM. **Lendo Botão com Serial monitor.** Disponível em: http://arduino.labdegaragem.com/Guia_preview/EK_4_botao_serial.html. Acesso em: 25 jun. 2020.

LANA, Hellynson Cássio; **Projetos Maker:** Arduino Eletrônica Robótica Automação residência. 1. ed. [S.I.]: Novatec Editora, 2018. p. 48-50.

LEITE & CORTE. **FALTA MÃO DE OBRA? ALUGUE UM ROBÔ PARA ORDENHAR AS VACAS E NÃO PERCA MAIS TEMPO!** Disponível em: <http://leitecorte.com.br/noticias/89/falta-m-o-de-obra->

alugue-um-rob-para-ordenhar-as-vacas-e-n-o-perca-mais-tempo. Acesso em: 16 jun. 2020.

MAKER PRO. **Learn how to use Tinkercad's newest feature, Codeblocks, to easily create intricate 3D designs.** Disponível em: <https://maker.pro/custom/tutorial/an-introduction-to-tinkercad-codeblocks>. Acesso em: 18 mai. 2020.

MASTER WALKER. **Arduino – Sequencial de LEDs com Potenciômetro.** Disponível em: <https://blogmasterwalkershop.com.br/arduino/arduino-sequencial-de-leds-com-potenciometro/>. Acesso em: 14 jul. 2020.

MASTER WALKER. **Como usar com Arduino – Controle Remoto Infravermelho.** Disponível em: <https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-controle-remoto-infravermelho/>. Acesso em: 6 ago. 2020.

MATARIC, Maia; **Introdução a Robótica.** 1. ed. São Paulo: Unesp, 2014. p. 21

MECAJUN. **Robótica e a exploração espacial.** Disponível em: <https://www.mecajun.com/2018/08/21/robotica-e-exploracao-espacial/>. Acesso em: 22 abr. 2020.

MEDIUM. **Simulando uma fechadura eletrônica controlada por senha.** Disponível em: <https://medium.com/@ludymilamelo00951/simulando-uma-fechadura-eletr%C3%B4nica-controlada-por-senha-ed625e6255b5>. Acesso em: 17 jul. 2020.

MICROBERTS, Michael; **Arduino Básico.** 2. ed. [S.I.]: Novatec, 2015. p. 20-32.

MIOTEC. **Robótica na Medicina: quais as principais inovações da área?** Disponível em: <https://blog.miotec.com.br/robotica-na-medicina/>. Acesso em: 13 ago. 2020.

MOBILE TIME. **Amazon começa teste com robô-entregador Scout.** Disponível em: <https://www.mobiletme.com.br/noticias/23/01/2019/amazon-comeca-teste-com-robo-entregador-scout/>. Acesso em: 5 jun. 2020.

MULTILÓGICA SHOP. **Arduino Guia Iniciante.** Disponível em: https://edisciplinas.usp.br/pluginfile.php/3252633/mod_resource/content/1/Guia_Arduino_Iniciante_Multilogica_Shop.pdf. Acesso em: 4 mai. 2020.

MUNDO DA ELÉTRICA. **Como medir corrente com multímetro?** Disponível em: <https://www.mundodaeletrica.com.br/como-medir-corrente-com-multimetro/>. Acesso em: 22 jun. 2020.

MUNDO DA ELÉTRICA. **Diodo retificador! O que é? Pra que serve?** Disponível em: <https://www.mundodaeletrica.com.br/diodo-retificador-o-que-e-pra-que-serve/?unapproved=64864&moderation-hash=a2a8b47ca0f7e0774c504731035d6a61#comment-64864>. Acesso em: 24 jun. 2020.

MUNDO DA ELÉTRICA. **O que é Servo motor e como funciona?** Disponível em: <https://www.mundodaeletrica.com.br/o-que-e-servo-motor-e-como-funciona/>. Acesso em: 30 jun. 2020.

MUNDO DA ELÉTRICA. **Tipos de capacitores.** Disponível em:

<https://www.mundodaeletrica.com.br/tipos-de-capacitores/>. Acesso em: 23 jun. 2020.

MUNDO PROJETADO. **Aprenda a controlar LEDs c/ o Arduino – Aula 3 – AI.** Disponível em: <http://mundoprojetado.com.br/aprenda-controlar-leds-c-o-arduino/>. Acesso em: 24 jun. 2020.

MUTCOM. **Instalando o Arduino IDE e configurando (Uno).** Disponível em: https://files.comunidades.net/mutcom/Instalando_o_Arduino_IDE_com_a_placa_UNO.pdf. Acesso em: 28 abr. 2020.

NASA. **Mars Rover Curiosity in Artist's Concept, Tall.** Disponível em: <https://www.jpl.nasa.gov/spaceimages/details.php?id=PIA14164>. Acesso em: 5 jun. 2020.

NTELLECTUALE TECNOLOGIA & TREINAMENTO. **O comando switch case em C.** Disponível em: <http://linguagemc.com.br/o-comando-switch-case-em-c/>. Acesso em: 22 mai. 2020.

NUSSEY, John; **Arduino Para Leigos:** subtítulo do livro. 2. ed. [S.I.]: Alta Books, 2019. p. 60-61.

OLHAR DIGITAL. **Carros e Tecnologia.** Disponível em: <https://olhardigital.com.br/carros-e-tecnologia/noticia/conheca-o-automower-um-robo-cortador-de-grama-que-esta-chegando-ao-brasil/79537>. Acesso em: 5 jun. 2020.

OLIVEIRA, C. L. V. et al. **Aprenda Arduino:** Uma abordagem prática. 1. ed. Brasil: [s.n.], 2018. p. 13-19.

OLIVEIRA, C. L. V. et al. **Arduino Básico:** subtítulo do livro. 1. ed. [S.I.]: Clube de Autores, 2018. p. 13-16.

OTTOBOCK. **Sistemas de Próteses de Membro Inferior.** Disponível em: <https://www.ottobock.com.br/prosthetics/membros-inferiores/solu%C3%A7%C3%B5es/>. Acesso em: 4 jun. 2020.

PARENT, Michel; LARGEAU, Claude; **Logic and Programming.** 4. ed. Nova York: Springer, 2014. p. 11

RENNA, R; PAIVA, L. **Tópicos Especiais em Eletrônica II Introdução ao microcontrolador Arduino Apostila de programação,** Niterói. 2014.

REPOSITÓRIO ROCA UTFPR. **DESENVOLVIMENTO DE UM ROBÔ PARALELO TIPO DELTA ASSOCIADO COM VISÃO COMPUTACIONAL PARA APLICAÇÕES PICK AND PLACE.** Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/929/1/CT_COMET_2012_1_02.pdf. Acesso em: 15 jun. 2020.

REPOSITORIO UFRN. **Desenvolvimento de uma cadeira de rodas robótica para o transporte de um portador de necessidades especiais.** Disponível em: https://repositorio.ufrn.br/jspui/bitstream/123456789/15476/1/IvoAON_DISSSERT.pdf. Acesso em: 15 jun. 2020.

RESEARCH GATE. **Modelagem da Cinemática Direta e Inversa do Robô SCARA EPSON G10-851S.** Disponível em: https://www.researchgate.net/figure/Figura-1-Robo-manipulador-SCARA-da-fabricante-EPSON_fig1_324088694. Acesso em: 8 jun. 2020.

ROBOCORE. Usando o Teclado Matricial com Arduino. Disponível em: <https://www.robocore.net/tutoriais/usando-teclado-matricial-com-arduino>. Acesso em: 9 jul. 2020.

ROBOCORE. Adicionando Bibliotecas na IDE Arduino. Disponível em: <https://www.robocore.net/tutoriais/adicionando-bibliotecas-na-ide-arduino>. Acesso em: 26 jun. 2020.

ROBOCORE. Como utilizar uma Protoboard. Disponível em: <https://www.robocore.net/tutoriais/como-utilizar-uma-protoboard>. Acesso em: 22 jun. 2020.

ROBOCORE. Termômetro com LCD (Junção de dois projetos). Disponível em: <https://www.robocore.net/modules.php?name=Forums&file=viewtopic&t=4507&newlang=english>. Acesso em: 25 jun. 2020.

ROBÓTICA. História da robótica. Disponível em: <http://roboticagrupo4.blogspot.com/2009/05/historia-da-robotica.html>. Acesso em: 8 abr. 2020.

SABER ELÉTRICA. Entenda o Funcionamento Tipos e Para Que Serve o Capacitor. Disponível em: <https://www.sabereletrica.com.br/entenda-o-funcionamento-dos-capacitores/>. Acesso em: 23 jun. 2020.

SCIHI BLOG. Jacques de Vaucanson and his Miraculous Automata. Disponível em: <http://scih.org/jacques-de-vaucanson-automata/>. Acesso em: 9 abr. 2020.

SHIGUEMORI, Ana Paula A. C. DISPLAY LCD: robótica educacional. Jacareí: Ifsp Jacareí, 2019. 15 slides, color.

SHIGUEMORI, Ana Paula A. C. ENTRADAS ANALÓGICAS: robótica educacional. Jacareí: Ifsp Jacareí, 2019. 65 slides, color.

SHIGUEMORI, Ana Paula A. C. PROJETO DE EXTENSÃO ROBOTGIRLS: robótica educacional. Jacareí: Ifsp Jacareí, 2019. 30 slides, color.

SHIGUEMORI, Ana Paula A. C. SENSOR DE UMIDADE E TEMPERATURA: robótica educacional. Jacareí: Ifsp Jacareí, 2019. 21 slides, color.

SHIGUEMORI, Ana Paula A. C. TECLADO KEYPAD: robótica educacional. Jacareí: Ifsp Jacareí, 2019. 21 slides, color.

SHIGUEMORI, Ana Paula A. C.; CRUZ, Ariadne Arrais. ENTRADA DIGITAL: robótica educacional. Jacareí: Ifsp Jacareí, 2019. 76 slides, color.

SHIGUEMORI, Ana Paula; ARRAIS, Ariadne. Introdução: robótica educacional. Jacareí: Ifsp Jacareí, 2019. 29 slides, color.

SHIGUEMORI, Ana Paula; Stekel, Tardelli. Linguagem C para Arduino. Jacareí: Ifsp Jacareí, 2020. 32 slides, color.

SHIGUEMORI, Ana Paula; Stekel, Tardelli. **Linguagem C para Arduino**. Jacareí: Ifsp Jacareí, 2020. 32 slides, color.

SOMBOX. **Como fazer a correta leitura de capacitores de cerâmico e poliéster**. Disponível em: <https://sombox.com.br/2014/01/01/como-fazer-a-correta-leitura-de-capacitores-de-ceramico-e-poliester/>. Acesso em: 22 jun. 2020.

SOUZA, F. **Robôs na Medicina**. [s.i]. 2005.

SOUZA, F. **Robôs não industriais**. [s.i]. 2005.

SQUIDS. **Jogo da Memória (Genius) - arduino jogo #03**. Disponível em: <http://www.squids.com.br/arduino/index.php/projetos-arduino/jogos/252-jogo-da-memoria-genius-arduino-jogo-03>. Acesso em: 29 jul. 2020.

SQUIDS. **Jogo das Palavras Embaralhadas - arduino jogo #02**. Disponível em: <http://www.squids.com.br/arduino/index.php/projetos-arduino/jogos/182-jogo-das-palavras-embaralhadas-arduino-jogo-02>. Acesso em: 16 jul. 2020.

SQUIDS. **Projeto 08 - Piano com buzzer**. Disponível em: <http://www.squids.com.br/arduino/index.php/projetos-arduino/projetos-squids/basico/100-projeto-08-piano-com-buzzer>. Acesso em: 15 jul. 2020.

SQUIDS. **Projeto 24 - Led RGB controlado por botões**. Disponível em: <http://www.squids.com.br/arduino/index.php/projetos-arduino/projetos-basicos/101-projeto-24-led-rgb-controlado-por-botoes>. Acesso em: 14 jul. 2020.

SQUIDS. **Projeto 30 - Sensor de luminosidade LDR com sinalizador de Leds**. Disponível em: <http://www.squids.com.br/arduino/index.php/projetos-arduino/projetos-basicos/115-projeto-29-sensor-de-luz-ldr-com-sinalizador-de-leds>. Acesso em: 14 jul. 2020.

SQUIDS. **Projeto 44 - Sensor de temperatura e umidade DHT11 com alarme RGB**. Disponível em: <http://www.squids.com.br/arduino/index.php/projetos-arduino/projetos-squids/basico/170-projeto-44-sensor-de-temperatura-e-umidade-com-alarme-rgb>. Acesso em: 29 jul. 2020.

SQUIDS. **Como usar push button com Arduino (programação)**. Disponível em: <http://www.squids.com.br/arduino/index.php/software/dicas/168-como-usar-push-button-com-arduino-programacao>. Acesso em: 20 jun. 2020.

THIZER . **Inteligência Artificial**. Disponível em: <https://www.thizer.com/blog/29-10-2018/inteligencia-artificial-uma-visao-geral.html>. Acesso em: 7 abr. 2020.

THIZER. **Inteligência Artificial**. Disponível em: <https://www.thizer.com/blog/29-10-2018/inteligencia-artificial-uma-visao-geral.html>. Acesso em: 3 jun. 2020.

Tópicos Especiais em Eletrônica II Introdução ao microcontrolador Arduino Apostila de programação

TRYBOTICS. **Temperatura Con Termistor NTC 10k Y Arduino**. Disponível em: <https://trybotics.com/project/Temperatura-con-Termistor-NTC-10k-y-Arduino-49660>. Acesso em: 7 jul. 2020.

UNIMATE. **The First Industrial Robot.** Disponível em: <https://www.robotics.org/joseph-engelberger/unimate.cfm>. Acesso em: 5 jun. 2020.

USINAINFO. **10 Projetos com Arduino para fazer em 2020.** Disponível em: <https://www.usinainfo.com.br/blog/10-projetos-com-arduino-para-fazer-em-2020/>. Acesso em: 4 ago. 2020.

USINAINFO. **Projeto Arduíno de irrigação automática.** Disponível em: <https://www.usinainfo.com.br/blog/projeto-arduino-de-irrigacao-automatica-sua-planta-sempre-bem-cuidada/>. Acesso em: 5 ago. 2020.

USINAINFO. **Projeto DHT11 Arduino medindo temperatura e umidade.** Disponível em: <https://www.usinainfo.com.br/blog/projeto-dht11-arduino-medindo-temperatura-e-umidade/>. Acesso em: 28 jul. 2020.

VEJA. **Cientistas fazem transplantes de braços bônicos controlados pela mente** Leia mais em: <https://veja.abril.com.br/ciencia/cientistas-fazem-transplantes-de-bracos-bonicos-controlados-pela-mente/>. Disponível em: <https://veja.abril.com.br/ciencia/cientistas-fazem-transplantes-de-bracos-bonicos-controlados-pela-mente/>. Acesso em: 5 jun. 2020.

VEJA. **MIT cria robô capaz de limpar vazamento de petróleo.** Disponível em: <https://veja.abril.com.br/ciencia/mit-cria-robo-capaz-de-limpar-vazamento-de-petroleo/>. Acesso em: 5 jun. 2020.

VIDA DE SILÍCIO. **Como utilizar o LED RGB.** Disponível em: <https://portal.vidadesilicio.com.br/como-utilizar-o-led-rgb-com-arduino/>. Acesso em: 6 jul. 2020.

VIDA DE SILÍCIO. **Robô seguidor de linha.** Disponível em: <https://portal.vidadesilicio.com.br/robo-seguidor-de-linha/>. Acesso em: 10 ago. 2020.

WARELINE TECNOLOGIA EM SAÚDE. **Entenda como a Robótica na Medicina vai revolucionar os procedimentos cirúrgicos.** Disponível em: <https://www.wareline.com.br/wareline/noticias/como-funciona-a-robotica-na-medicina/>. Acesso em: 15 abr. 2020.

WARREN, John-David; ADAMS, Josh; MOLLE, Harald; **Arduino para robótica.** 1. ed. [S.l.]: Blucher, 2019. p. 15-20.

WARREN, John-David; ADAMS, Josh; MOLLE, Harald; **Arduino para robótica.** 1. ed. [S.l.]: Blucher, 2019. p. 15-20.

WIKIPÉDIA. **Eolípila.** Disponível em: <https://pt.wikipedia.org/wiki/Eol%C3%ADpila>. Acesso em: 3 jun. 2020.

WIKIPÉDIA. **Joseph Engelberger.** Disponível em: https://en.wikipedia.org/wiki/Joseph_Engelberger. Acesso em: 4 jun. 2020.

WIKIPÉDIA. **General Atomics MQ-9 Reaper.** Disponível em: https://pt.wikipedia.org/wiki/General_Atomics_MQ-9_Reaper. Acesso em: 4 jun. 2020.

WIKIPÉDIA. **Lunokhod 1.** Disponível em: https://pt.wikipedia.org/wiki/Lunokhod_1. Acesso em: 4

jun. 2020.

WIKIPÉDIA. **Yutu**. Disponível em: <https://pt.wikipedia.org/wiki/Yutu>. Acesso em: 4 jun. 2020.

WIKIWAND. **Jacques de Vaucanson**. Disponível em: https://www.wikiwand.com/en/Jacques_de_Vaucanson. Acesso em: 4 jun. 2020.

WORLD OBSERVER BY CLAUDIA & VICENTE. **Há dois Mil Anos Atrás – O Genial Heron de Alexandria**. Disponível em: <https://vicentemanera.com/2013/12/03/ha-dois-mil-anos-atras-o-genial-heron-de-alexandria/>. Acesso em: 2 jun. 2020.