

epb

escola profissional
de braga

PAP - PROVA DE APTIDÃO PROFISSIONAL



Casa e Garagem em Arduino

CURSO PROFISSIONAL De Gestão e
Programação de Sistemas
Informáticos

Nuno Miguel Amaral da Costa

Trabalho realizado sob a orientação de:

Prof. Manuel Ferreira

Prof. Ana Marta Vilaverde

Escola Profissional de Braga

Braga, 10 de maio de 2024

Nota Introdutória

A Prova de Aptidão Profissional consiste na apresentação e defesa, perante um júri, de um projeto, consubstanciado num produto, material ou intelectual, numa intervenção ou numa atuação, bem como do respetivo relatório final de realização e apreciação crítica, demonstrativo de saberes e competências profissionais, adquiridos ao longo da formação e estruturante do futuro profissional.

A sua realização constitui-se como um dos imperativos legais para a conclusão do curso profissional de (designação do curso), que confere uma dupla certificação: qualificação profissional de nível IV e o 12º ano como certificação escolar de nível secundário.

A presente prova foi realizada na Escola Profissional de Braga no presente ano letivo 2023/2024 e a sua defesa realizada, perante um júri final, nas datas estabelecidas no calendário escolar.

2 Nuno Miguel Amaral da Costa



Os Fundos Europeus mais próximos de si.



Cofinanciado pela
União Europeia



uma escola

Rumos
education

Knowledge
sharing

Dedicatória

Dedico está Pap ao meu Pai Fernando de Sousa Pires que me ensinou o básico sobre hardware entre outras várias coisas e a minha mãe que me incentivou e ajudou a fazer a minha Pap.

Nuno Miguel Amaral da Costa

3



uma escola



Agradecimentos

Agradeço ao Prof. Manuel Ferreira que me deu a ideia de fazer esta Pap e ajudou a fazer a mesma e aos outros professores que me ajudaram muito.

4

Nuno Miguel Amaral da Costa



Cofinanciado pela
União Europeia



uma escola

Rumos
education

Knowledge
sharing

Índice

1. Introdução.....	8
1.1. Programas.....	10
1.2. O que é Arduíno.....	10
1.3. Arduino IDE.....	11
1.4. Tinkercad	13
1.5. OneDrive	14
1.6 Youtube	15
1.7 Arduino Uno R3 Original.....	16
1.8 Arduino Uno Rev3.....	18
1.9 Arduino Mega 2560 R3 Keystudio.....	20
1.10 BreadBoard.....	21
1.11 Sensor HC-SR04 (Sensor de Movimento).....	22
1.12 Sensor de Temperatura e Humidade Cm232.....	24
1.13 Display Lcd 16x2 Retroiluminacao Amarela.....	25
1.14 LEDs.....	27
1.15 Resistências.....	28

2.

2.1 Fotos de Arduinos e Códigos dos Testes:

1. Teste1.....	30
2. Teste2.....	32

2.2 Arduinos e códigos Finais:

1. Arduino Do Lcd 16x2 e sensor de Humidade.....	34
2. Arduino Sensor de Movimento e Servo Motor.....	37

2.3 Maquete da Casa e Garagem com o Arduino dos sensores de movimento e servo motor incluído.....

41

2.4 Maquete da Casa e Garagem Projeto Final Com o Arduino dos sensores de movimento e servo motor incluído e o Arduino Do Lcd 16x2 e sensor de Humidade incluído.....

42

Conclusão.....

43

Bibliografia.....

44

Anexos.....

45

Índice de Figuras

Figura 1 -C++..... 6



Introdução

No meu trabalho vou fazer um sistema Arduino (uma garagem e uma casa).

Vou fazer uma maquete de uma garagem com uma casa ao lado.

A maquete terá um sensor de presença para abrir a garagem, um motor para a porta e um sensor de proximidade para fazer com que a porta se abra.

Na casa vou utilizar 5 leds para acender as luzes.

Vou utilizar um display para dar informações como a temperatura entre outras.

1. Reforçar a capacidade de superar situações inesperadas ou imprevistas nas diferentes fases do projeto;
2. Desenvolver o espírito crítico e sentido de autonomia;
3. Aplicar conhecimentos e competências adquiridos nas diferentes disciplinas;
4. Aumentar a minha capacidade de domínio sobre este tema;
5. Aprender sobre Arduino e as suas funcionalidades.

Cronograma		
Fase	Calendarização	Elenco de atividades nas diversas fases
Conceção	Setembro 2023	Conhecimento do regulamento
	Setembro 2023	Definição dos critérios da PAP pela direção
	Setembro 2023	Período de negociação / reflexão / decisão sobre os temas / problemas a explorar
	Setembro 2023	Normas para a elaboração de um trabalho científico
	Até 31 outubro 2023	Entrega do projeto de acordo com as normas do regulamento
Desenvolvimento	Novembro 2023	Teste dos diversos componentes e da placa
	Novembro/Dezembro 2023	Criação do código no Thinkercad
	Janeiro 2024/Março	Montagem da maquete
	Abril 2024	Testes e correções
Autoavaliação e Relatório	Janeiro 2024	Defesa intermédia da PAP
	6 maio 2024	Conclusão do projeto e entrega do relatório
	Julho 2024	Defesa final da PAP

1.1 Programas

Na minha PAP utilizei três Arduino em que a linguagem de programação é C/C++.

Programas utilizados:

Arduino

Arduino IDE

Tinkercad

OneDrive

Youtube

1.2 O que é o Arduino



O Arduino foi criado por um grupo de cinco membros: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. A ideia inicial era desenvolver uma plataforma de prototipagem eletrónica fácil de usar e acessível para artistas, designers, entusiastas e qualquer pessoa interessada em criar projetos interativos.

A história do Arduino começou em 2003, quando Massimo Banzi e David Cuartirllles, inspirados em projetos de hardware aberto, decidiram criar uma plataforma que tornasse a eletrónica mais

10 Nuno Miguel Amaral da Costa

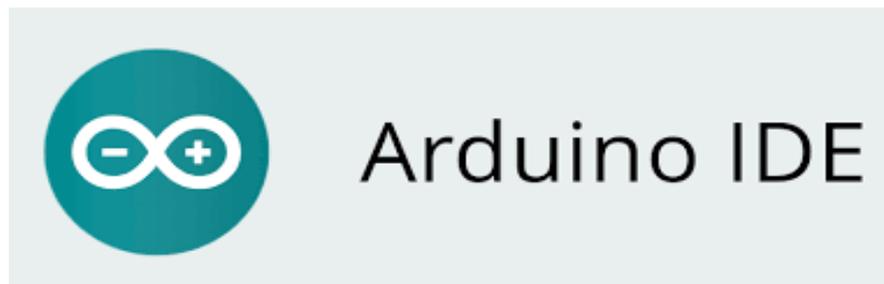
acessível. Em 2005, lançaram a primeira placa Arduino, conhecida como Arduino Diecimila, marcando oficialmente o início da plataforma Arduino.

O Arduino foi projetado para ser uma ferramenta de prototipagem rápida e fácil, permitindo que os utilizadores testem e implementem suas ideias sem a necessidade de profundo conhecimento em eletrónica ou programação. Desde então, a família de placas Arduino cresceu, oferecendo diferentes modelos com recursos variados para atender a diversas necessidades de projetos.

A plataforma Arduino também se destacou pela sua abordagem de código aberto. O hardware e o software Arduino são disponibilizados com licenças de código aberto, encorajando a comunidade a contribuir, modificar e compartilhar melhorias.

Ao longo dos anos, o Arduino ganhou enorme popularidade e tornou-se uma ferramenta fundamental para educadores, estudantes, engenheiros, artistas e entusiastas em todo o mundo, desempenhando um papel significativo na promoção da cultura da eletrónica de código aberto.

1.3 Arduino IDE



A aplicação Arduino IDE (Integrated Development Environment ou Ambiente de Desenvolvimento Integrado, em português) é um software usado para programar placas Arduino. Esta aplicação oferece uma interface de programação que permite aos utilizadores escrever, compilar e fazer o upload de código para o microcontrolador presente nas placas Arduino.

Algumas características principais da Arduino IDE incluem:

Editor de Código: A IDE possui um editor de código no qual os utilizadores podem escrever o seu programa em uma linguagem baseada em C/C++ simplificada para facilitar a programação.

Compilador: A Arduino IDE inclui um compilador que traduz o código-fonte escrito pelo utilizador para instruções compreensíveis pelo microcontrolador da placa Arduino.

Monitor Serial: A IDE permite a comunicação serial com placa Arduino, o que é útil para depuração e para visualizar mensagens ou dados que são enviados ou recebidos pela placa.

Gerenciador de Bibliotecas: Facilita a instalação e a gestão de bibliotecas de software, que são conjuntos de código pré-escrito que podem ser utilizados para realizar tarefas específicas.

Ferramentas de Upload: A IDE permite que os utilizadores carreguem o código compilado para a placa Arduino através de uma conexão USB.

A Arduino IDE é uma ferramenta essencial para quem trabalha com placas Arduino, tornando o processo de processo de programação e desenvolvimento de projetos eletrónicos mais acessível, especialmente para iniciantes. No entanto, é importante notar que, ao longo do tempo, surgiram outras opções de ambientes de desenvolvimento para Arduino, oferecendo funcionalidades adicionais e interfaces aprimoradas.

1.4 Tinkercad



AUTODESK®
TINKERCAD®

12 Nuno Miguel Amaral da Costa

O Tinkercad é uma plataforma online de design 3D que permite aos utilizadores criar modelos tridimensionais de forma fácil e intuitiva. Desenvolvido pela Autodesk, o Tinkercad é projetado para ser acessível a iniciantes e estudantes, oferecendo uma interface simples e ferramentas básicas de modelagem.

A plataforma é frequentemente usada em ambientes educacionais para ensinar conceitos de design 3D e modelagem para crianças e adolescentes. Com o Tinkercad, os utilizadores podem criar objetos virtuais tridimensionais manipulando formas básicas, combinando-as, cortando-as e personalizando-as conforme necessário.

O Tinkercad, apesar de ser conhecido principalmente pela sua funcionalidade de design 3D, também oferece uma secção dedicada aos “Circuitos”, onde os utilizadores podem projetar e simular circuitos eletrónicos. Nesta parte do Tinkercad, os utilizadores podem explorar e aprender sobre eletrónica, programação e interações entre os componentes.

A plataforma de Circuitos do Tinkercad disponibiliza uma biblioteca extensa de componentes eletrónicos virtuais, como resistores, transístores, LEDs, microcontroladores e sensores. Os utilizadores podem arrastar e soltar esses componentes numa área de trabalho virtual e, em seguida, conectar os fios para criar circuitos eletrónicos.

Além disso, a plataforma permite que os utilizadores programem os microcontroladores usando linguagem de programação visual, simplificando a aprendizagem de programação para aqueles que estão a começar. A capacidade de simular o comportamento dos circuitos antes da implementação física é uma característica valiosa para aprender sobre eletrónica sem a necessidade de hardware real.

Esta funcionalidade de hardware no Tinkercad é útil para estudantes, entusiastas e curiosos de eletrónica que desejam experimentar com circuitos virtuais, testar as suas ideias e compreender os princípios básicos da eletrónica de uma forma prática e interativa.

1.5 OneDrive



O OneDrive é um serviço de armazenamento na nuvem desenvolvido pela Microsoft. Este serviço permite que os utilizadores armazenem e acessem a ficheiros remotamente a partir de vários dispositivos. O OneDrive oferece a capacidade de sincronizar ficheiros e pastas entre computadores, dispositivos móveis e a nuvem, facilitando o acesso aos dados em qualquer lugar.

O OneDrive foi lançado inicialmente em agosto de 2007 como “Windows Live Folders” e, posteriormente, foi renomeado para “Windows Live SkyDrive”. Em 2013, o serviço foi rebatizado como OneDrive devido a uma disputa legal sobre o nome SkyDrive.

A Microsoft continuou a melhorar o OneDrive ao longo dos anos, integrando-o com outros produtos e serviços, como o Windows 10 e o pacote Office 365. O serviço oferece opções tanto gratuitas como planos pagos com mais capacidade de armazenamento e funcionalidades adicionais.

O OneDrive é uma parte fundamental do ecossistema da Microsoft, proporcionando aos utilizadores uma solução conveniente para armazenamento na nuvem e colaboração em documentos e ficheiros.

1.6 YouTube



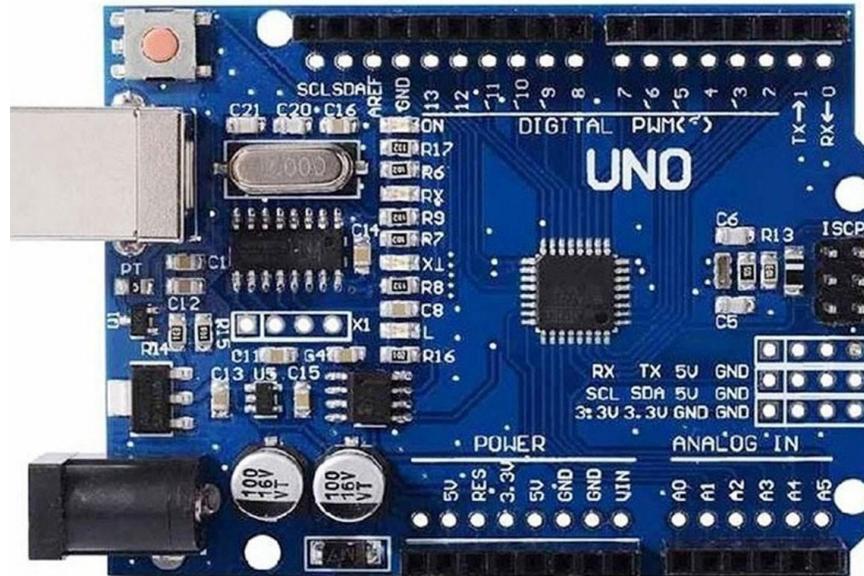
14 Nuno Miguel Amaral da Costa

O YouTube é uma plataforma de partilha de vídeos online. Permite que os utilizadores carreguem, vejam, comentem e partilhem vídeos. O YouTube foi criado por três funcionários do Paypal: Chad Hurley, Steve Chen e Jawed Karim. Eles fundaram a plataforma em fevereiro de 2005 e, no mesmo ano, o YouTube foi oficialmente lançado publicamente em novembro.

O conceito inicial do YouTube era fornecer um espaço online onde as pessoas pudessem partilhar vídeos facilmente. O site cresceu rapidamente em popularidade devido à sua abordagem intuitiva e à capacidade dos utilizadores de encontrarem e partilharem uma ampla variedade de conteúdos, desde vídeos caseiros até produções mais profissionais.

Em 2006, o YouTube foi adquirido pelo Google, o que ajudou a impulsionar ainda mais o seu crescimento e integração com outras plataformas e serviços online. Atualmente, o YouTube é uma das maiores plataformas de partilha de vídeos do mundo, oferecendo uma vasta gama de conteúdos, desde vídeos educativos até entretenimento, música, vlogs e muito mais.

1.7 Arduino Uno R3 Original



O Arduino Uno R3 é uma das placas de desenvolvimento mais conhecidas e utilizadas da família Arduino. Aqui estão algumas características e informações sobre o Arduino Uno R3:

Microcontrolador: Utiliza o microcontrolador ATmega328P da Atmel, um microcontrolador AVR de 8 bits.

Portas de Entrada/Saída(I/O): Possui 14 pinos digitais de I/O, dos quais 6 podem ser usados como saídas PWM (Modulação por Largura de Pulso) e 6 como entradas analógicas.

Memória: Conta com 32 KB de memória flash para armazenamento de código, 2KB de SRAM e 1 KB de EEPROM.

Interfaces: Inclui uma porta USB (geralmente USB-B) para programação e comunicação com o computador, um conector de alimentação DC, conectores de entrada/saída, um botão de reset e outros.

Compatibilidade com Shields: É compatível com uma variedade de shields (módulos de expansão) Arduino, que podem ser conectados diretamente à placa para adicionar funcionalidades específicas, como sensores, módulos de comunicação, entre outros.

Alimentação: Pode ser alimentado por meio da conexão USB, uma fonte de alimentação externa ou uma bateria. Opera com uma faixa de voltagem de 7 a 12 volts.

Evolução R3: A versão R3 do Arduino Uno trouxe algumas melhorias em relação às versões anteriores, incluindo a adição de pinos DAS e SCL perto do AREF, e um novo Atmega16U2 no lugar do antigo Atmega8U2 para a interface USB.

Software: Programado usando a linguagem de programação Arduino, que é baseada em C/C++. O software Arduino IDE é usado para escrever, compilar e enviar código para a placa.

O Arduino Uno R3 é amplamente utilizado em projetos de eletrónica, robótica, automação residencial, aprendizagem de programação e muitos outros campos devido à sua simplicidade, versatilidade e robustez.

1.8 Arduino Uno Rev3



O Arduino Uno Rev3 é uma versão específica da placa de desenvolvimento Arduino Uno, e “Rev3” refere-se à revisão mais recente da placa . Aqui estão algumas características e informações sobre o Arduino Uno Rev3:

Microcontrolador: Utiliza o microcontrolador ATmega328P da Atmel, um microcontrolador AVR de 8 bits.

Portas de Entrada/Saída(I/O): Possui 14 pinos digitais de I/O, dos quais 6 podem ser usados como saídas PWM (Modulação por Largura de Pulso) e 6 como entradas analógicas.

Memória: Conta com 32 KB de memória flash para armazenamento de código, 2KB de SRAM e 1 KB de EEPROM.

Interfaces: Inclui uma porta USB (geralmente USB-B) para programação e comunicação com o computador, um conector de alimentação DC, conectores de entrada/saída, um botão de reset e outros.

Compatibilidade com Shields: É compatível com uma variedade de shields (módulos de expansão) Arduino, que podem ser conectados diretamente à placa para adicionar funcionalidades específicas, como sensores, módulos de comunicação, entre outros.

Alimentação: Pode ser alimentado por meio da conexão USB, uma fonte de alimentação externa ou uma bateria. Opera com uma faixa de voltagem de 7 a 12 volts.

Comunicação Série: Além da comunicação USB para programação, o Arduino Uno Rev3 suporta comunicação serial através dos pinos RX e TX, permitindo a interação com outros dispositivos.

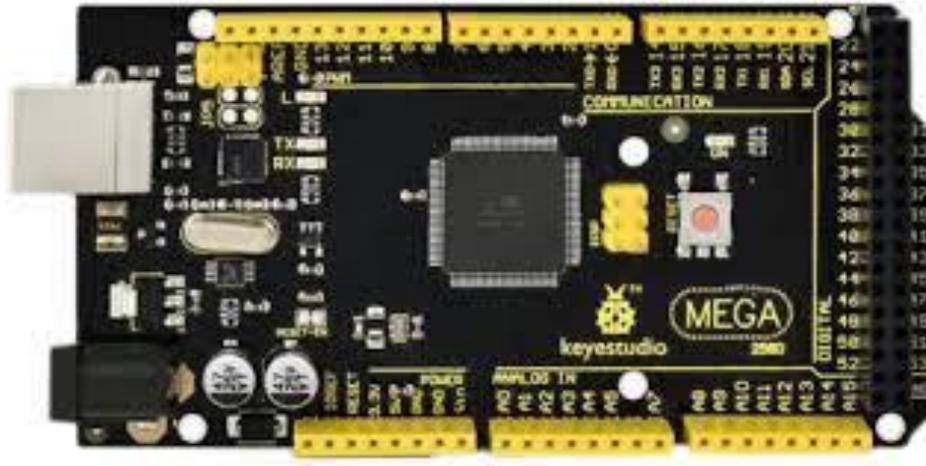
Evolução R3: A versão R3 introduziu algumas melhorias em relação às versões anteriores, incluindo a adição de pinos DAS e SCL perto do AREF, e a substituição do chip de interface USB para uma versão Atmega16U2

Software: Programado usando a linguagem de programação Arduino, que é baseada em C/C++. O software Arduino IDE é usado para escrever, compilar e enviar código para a placa.

O Arduino Uno R3 é uma escolha comum para projetos de eletrónica, robótica, automação residencial e educação, devido à sua facilidade de uso, documentação extensiva e suporte da comunidade.

1.9 Arduino Mega 2560 R3 Keystudio

Nuno Miguel Amaral da Costa 19



O Arduino Mega 2560 é uma versão avançada da família Arduino, sendo que o modelo R3 refere-se à revisão mais recente da placa. No entanto, é importante esclarecer que “Keyestudio” é uma marca de fabricante específico que produz placas e kits compatíveis com Arduino. Assim sendo, o “Arduino Mega 2560 R3 Keyestudio” seria uma versão do Arduino Mega 2560 produzida ou comercializada pela Keyestudio. Aqui estão algumas informações gerais sobre o Arduino Mega 2560 R3:

Microcontrolador: Utiliza o microcontrolador Atmega2560 da Atmel, que é um microcontrolador AVR de 8 bits com maior capacidade em comparação com o Atmega328P usado no Arduino Uno.

Portas de Entrada/Saída(I/O): Possui 54 pinos digitais de I/O, incluindo 15 pinos que podem ser usados como saídas PWM, e 16 entradas analógicas.

Memória: Conta com 256 KB de memória flash para armazenamento de código, 8 KB de SRAM e 4KB de EEPROM.

Interfaces: Inclui múltiplas portas USB, conectores de alimentação, entradas/saídas digitais e analógicas, e interfaces de comunicação, como UART, I2C e SPI.

Compatibilidade com Shields: Pode ser usado como uma variedade de shields Arduino projetados para encaixar diretamente na placa e expandir suas capacidades.

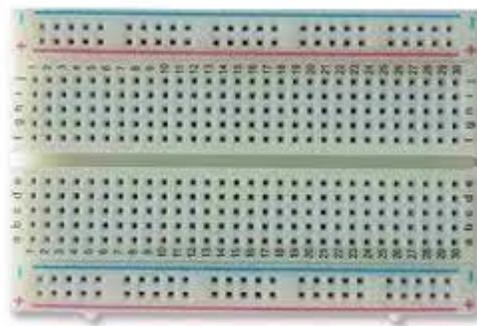
Alimentação: Pode ser alimentado por meio de uma fonte de alimentação externa ou via USB. A faixa de voltagem recomendada é de 7 a 12 volts.

Comunicação Serial: Oferece comunicação serial através de múltiplos pinos UART (RX/TX).

Software: Programando utilizando a linguagem de programação Arduino, baseada em C/C++. O software Arduino IDE é usado para escrever, compilar e enviar código para a placa.

Uso Comum: O Arduino Mega 2560 é frequentemente escolhido para projetos que exigem mais pinos de I/O, memória e capacidade de processamento do que o Arduino Uno.

1.10 BreadBoard



A BreadBoard, também conhecida como placa de ensaio ou placa de prototipagem, é uma ferramenta essencial para projetos eletrónicos e prototipagem rápida. Fornece uma plataforma na qual os componentes eletrónicos podem ser temporariamente montados sem a necessidade de soldadura. Essa flexibilidade permite que os engenheiros, estudantes e entusiastas testem e modifiquem circuitos de forma fácil e rápida.

Aqui estão algumas características e informações sobre BreadBoards:

Design: A BreadBoard possui uma matriz de furos conectados eletricamente internamente. Os furos são organizados em linhas e colunas, facilitando a conexão de componentes.

Conexões Internas: Cada linha vertical de furos está conectada internamente, enquanto os furos ao longo das linhas horizontais são conectados entre si. Isso permite que os utilizadores conectem componentes, como resistores, LEDs, fios e circuitos integrados, de maneira rápida e intuitiva.

Alimentação: Geralmente, as BreadBoards têm duas seções distintas para alimentação. Uma seção é para a alimentação positiva (+) e a outra para a alimentação negativa (-), facilitando a conexão de uma fonte de alimentação ao circuito.

Tamanhos: Existem BreadBoards de diferentes tamanhos, desde pequenas para projetos mais complexos. Algumas BreadBoards podem ser interligadas para criar áreas de trabalho maiores.

Linhas Dados e Alimentação: Além das linhas de dados e alimentação, muitas BreadBoards têm áreas designadas para conectores IC (circuitos integrados) e para a montagem de componentes específicos, como microcontroladores.

Durabilidade: As BreadBoards são reutilizáveis e duráveis, permitindo que os utilizadores montem e desmontem circuitos várias vezes sem causar danos.

Facilidade de Uso: A BreadBoard é uma ferramenta ideal para aprendizes, pois permite a prototipagem rápida sem a necessidade de equipamento de soldagem. Isso facilita experimentação e a compreensão dos princípios básicos de eletrónica.

1.11 Sensor HC-SR04 (Sensor de Movimento)



O Sensor HC-SR04 é um sensor ultrassónico utilizado para medir distâncias. É bastante popular em projetos de eletrónica e robótica devido à sua facilidade de utilização e custo acessível. Aqui estão algumas características e informações sobre o sensor HC-SR04:

Princípio de Funcionamento: O HC-SR04 utiliza ondas ultrassónicas para medir distâncias. Emite um sinal ultrassónico através de um transmissor e aguarda o eco do sinal refletido por um objeto. A medida do tempo que o sinal leva para viajar até ao objeto e retornar permite calcular a distância.

Especificações Técnicas:

- **Alcance:** O alcance típico do HC-SR04 varia de 2 cm a 4 metros.
- **Precisão:** A precisão pode ser afetada por condições ambientais, como temperatura e humidade.
- **Ângulo de Detecção:** O sensor opera num ângulo estreito, sendo eficaz para medições frontais.

Pinos do Sensor:

- **VCC:** Alimentação (normalmente entre 5V e 12V).
- **Trig(Trigger):** Pino de entrada para iniciar a medição.
- **Echo:** Pino de saída que fornece um pulso cuja duração é proporcional à distância medida.
- **GND:** Terra (0V).

Modo de Operação:

- O Utilizador envia um pulso curto ao pino Trig para iniciar a medição.
- O sensor emite um sinal ultrassónico e aguarda o retorno do eco.

- *O pino Echo fornece um pulso cuja largura está relacionada à distância medida.*
- *Utilizando a medida do tempo de viagem, a distância pode ser calculada.*

Programação:

- *A leitura da distância geralmente é feita utilizando uma placa de desenvolvimento como Arduino.*
- *O código Arduino pode medir o tempo de duração do pulso no pino Echo e converter isso em uma distância utilizando a fórmula adequada.*

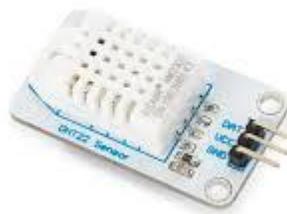
Aplicações Comuns:

- *Evitar obstáculos em robótica.*
- *Medição de distâncias em projetos de automação.*
- *Controlo de presença em projetos de seguranças.*

Considerações de Uso:

- *O sensor opera melhor em superfícies que refletem bem o som, como objetos sólidos e planos.*
- *Condições ambientais, como a presença de outros sons ultrassónicos, podem afetar o desempenho.*

1.12 Sensor de Temperatura e Humidade Cm232



Sensores de temperatura e humidade geralmente seguem princípios semelhantes. Eles são projetados para medir a temperatura e a humidade relativa do ambiente em que estão instalados. Aqui estão algumas informações gerais sobre sensores de temperatura e humidade para lhe dar uma ideia do que esses dispositivos fazem:

Nuno Miguel Amaral da Costa 23

Princípio de Funcionamento: Sensores de temperatura e humidade usam diferentes tecnologias para medir as condições ambientais. Alguns utilizam termístores para medir a temperatura, enquanto outros usam sensores capacitivos para medir a humidade relativa.

Leituras: Eles fornecem leituras em unidades como graus Celsius (para temperatura) e percentagem (para humidade relativa).

Calibração: Alguns sensores são calibrados de fábrica, enquanto outros podem precisar de calibração periódica para garantir precisão nas leituras.

Interface: Alguns sensores têm interfaces digitais, como I2C ou UART, facilitando a integração com microcontroladores e sistemas embarcados.

Alimentação: Geralmente, esses sensores querem uma fonte de alimentação específica, e alguns modelos operam em baixa voltagem para maior eficiência energética.

Aplicações Comuns: Sensores de temperatura e humidade são usados em uma variedade de aplicações, incluindo monitoramento ambiental, controle de clima em ambientes fechados, sistemas de aquecimento, ventilação e ar condicionado (HVAC); e projetos DIY.

1.13 Display Lcd 16x2 Retroiluminacao Amarela



Um Display LCD 16x2 com Retroiluminação Amarela refere-se a um tipo de ecrã de cristal líquido que tem a capacidade de exibir 16 caracteres em 2 linhas e possui uma luz de fundo (retroiluminação) na

cor amarela. Este tipo de ecrã é comumente utilizado em projetos eletrónicos, sistemas embarcados e prototipagem.

Aqui estão algumas características e informações sobre o Display LCD 16x2 com Retroiluminação Amarela:

Formato de Exibição: O “16x2” indica que o ecrã tem a capacidade de mostrar 16 caracteres em cada uma das 2 linhas, totalizando 32 caracteres visíveis simultaneamente.

Retroiluminação Amarela: A presença de retroiluminação amarela significa que o ecrã possui LEDs amarelos que iluminam o fundo, facilitando a leitura do conteúdo em condições de pouca luz.

Modo de Operação: Os ecrãs LCD 16x2 são frequentemente utilizados em modo paralelo ou modo série (usando um módulo de interface).

Controlador: O ecrã LCD 16x2 geralmente utiliza um controlador compatível com HD44780, que é amplamente utilizado e suportado, facilitando a programação e o uso com diferentes plataformas.

Interface: O ecrã pode ser controlado por microcontroladores, como Arduino, Raspberry Pi, ou outros dispositivos eletrónicos. Geralmente, requer a utilização de pinos para controle de dados e controle de sinal.

Aplicações Comuns: Ecrãs LCD são utilizados numa variedade de projetos, incluindo medidores, sistemas de controlo, termostatos, relógios, e outros dispositivos onde exibição de informações é necessária.

Programação: Para controlar o ecrã, os programadores geralmente utilizam bibliotecas específicas para o microcontrolador ou plataforma que estão usando. No caso do Arduino, por exemplo, a biblioteca “LiquidCrystal” é frequentemente empregada.

Alimentação: O ecrã pode requerer uma fonte de alimentação específica, geralmente 5V, para operar corretamente.

1.14 LEDs



Nuno Miguel Amaral da Costa 25

Os LEDs (Díodos Emissores de Luz) são componentes essenciais em projetos eletrónicos, incluindo aqueles desenvolvidos com placas Arduino. Aqui estão algumas informações sobre LEDs no contexto do Arduino:

O que é um LED: Um LED é um dispositivo semicondutor que emite luz quando uma corrente elétrica passa por ele. A cor da luz emitida depende do material semicondutor utilizado no LED.

Tipos de LEDs:

- **LEDs de Cor Única:** *Imitem luz de uma única cor, como vermelho, verde ou azul.*
- **LEDs RGB (Red, Green, Blue):** *Permite a mistura de cores para produzir uma variedade de tons.*
- **LEDs Bicolores e Tricolores:** *Podem exibir duas ou três cores, dependendo da polarização.*

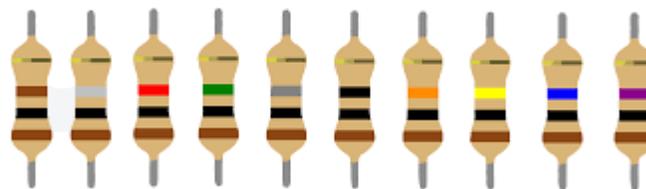
Conexão ao Arduino: Os LEDs são frequentemente conectados a pinos digitais no Arduino. O terminal mais longo do LED (ânodo) geralmente é conectado ao pino digital, enquanto o terminal mais curto (cátodo) é conectado ao terra (GND). A resistência é muitas vezes adicionada ao circuito para limitar a corrente e evitar danos ao LED.

Controle de LEDs com Arduino: A biblioteca 'digitalWrite()' no Arduino é frequentemente usada para ligar ou desligar um LED. Pode-se usar a função 'analogWrite' para controlar a intensidade luminosa de LEDs PWM (Pulse Width Modulation). Bibliotecas específicas, como a 'FastLED' para LEDs RGB, são utilizadas para um controlo mais avançado.

Uso em Projetos: Os LEDs são utilizados para indicar estados, sinalizar eventos, criar efeitos visuais e fornecer feedback visual em muitos projetos Arduino.

Cuidados: Ao usar LEDs, é importante garantir que a corrente e a tensão aplicadas estejam dentro dos limites recomendados pelo datasheet do LED.

1.15 Resistências



O que são Resistências: As resistências são componentes eletrónicas que têm a propriedade de resistir ao fluxo de corrente elétrica. São representadas em ohms (Ω).

Função no Arduino: As resistências são usadas para limitar a corrente que flui através de outros componentes, protegendo-os de danos causados por correntes excessivas.

Tipos de Resistências:

- **Resistências de Filme Fixo:** São as mais comuns e vêm em valores de resistência específicos.
- **Resistências de Potências:** Projetadas para lidar com correntes mais elevadas, são usadas em situações onde a dissipação de calor é um fator crítico.
- **Resistências Variáveis (Potenciômetros):** Permitem ajustar manualmente a resistência.

Código de Cores:

- As resistências de filme fixo geralmente têm um código de cores que indica o valor da resistência. Os quatro ou cinco anéis coloridos na resistência representam dígitos numérico e a tolerância.

Cálculos de Resistência:

- A Lei de Ohm ($V = I * R$) é usada para calcular a resistência (R) com base na tensão (V) e na corrente (I).
- Para calcular a corrente, pode-se usar a Lei de Ohm como $I = V / R$.

Uso com LEDs:

- Ao usar LEDs com placas Arduino, é comum adicionar uma resistência em série para limitar a corrente e evitar danos ao LED.

Proteção de Portas do Arduino:

- Resistências são frequentemente usadas para proteger as portas de entrada/saída do Arduino contra correntes excessivas que podem danificar os pinos.

Conexão:

- As resistências são conectadas em série com outros componentes no circuito. A extremidade da resistência pode ser conectada a um pino digital ou analógico, enquanto a outra extremidade está conectada ao terra (GND) ou ao positivo (VCC).

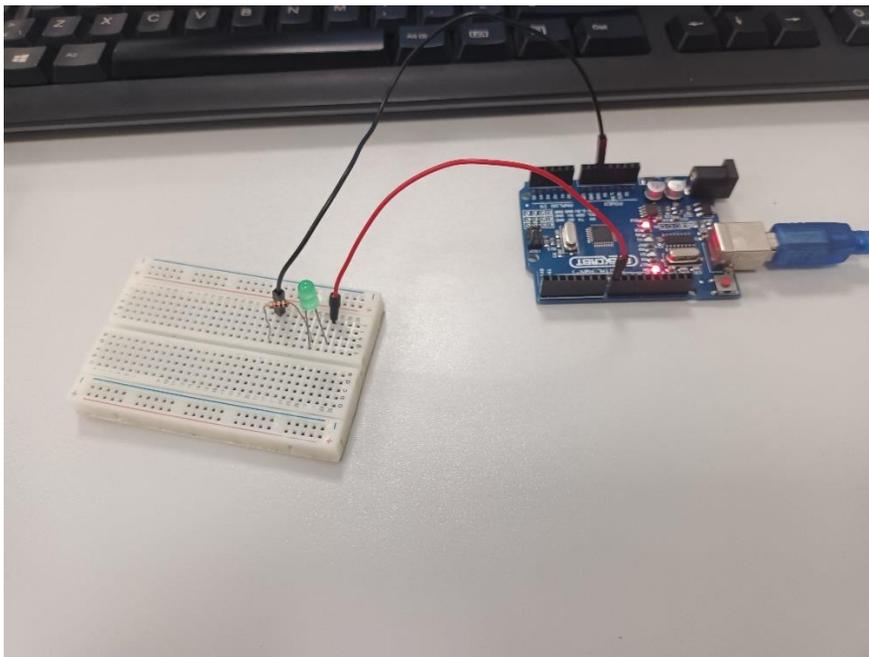
Valor de Resistências Padrão:

- Resistências com valores padronizados (como 220Ω, 1kΩ, 10kΩ) são comumente usadas em projetos Arduino.

2.

2.1 Fotos de Arduinos e Códigos dos Testes:

Teste1



28 Nuno Miguel Amaral da Costa

```
int LED = 8;
void setup() {
  pinMode(LED, OUTPUT);
}

void loop() {
  digitalWrite(LED, HIGH);
  delay(500);
  digitalWrite(LED, LOW);
  delay(500);
}
```

Este teste é um exemplo básico para piscar um LED que está conectado ao pino 8 do Arduino, usei este teste para saber quais LEDs e resistências estavam a funcionar.

Agora vou explicar o código:

“int LED =8;” – Esta linha declara uma variável ‘LED’ do tipo inteiro e atribui o valor 8 a ela. Isso significa que o pino 8 do Arduino será usado para controlar o LED.

“void setup() { pinMode(LED, OUTPUT); }” – o método ‘setup()’ é chamado uma vez quando o Arduino é inicializado. Neste caso, ele define o pino ‘LED’ como saída, usado a função ‘pinMode()’. Isso significa que o Arduino enviará sinais de controle para este pino.

“void loop() { digitalWrite(LED, HIGH); delay(500); digitalWrite(LED, LOW); delay(500); }” – O método ‘loop()’ é executado continuamente após a inicialização. Dentro dele, o LED é ligado (configurando o pino ‘LED’ como HIGH) por 500 milissegundos (meio segundo), depois desligado (configurando o pino ‘LED’ como LOW) por mais 500 milissegundos. Este ciclo continua indefinidamente, fazendo o LED piscar.

Conexão dos cabos:

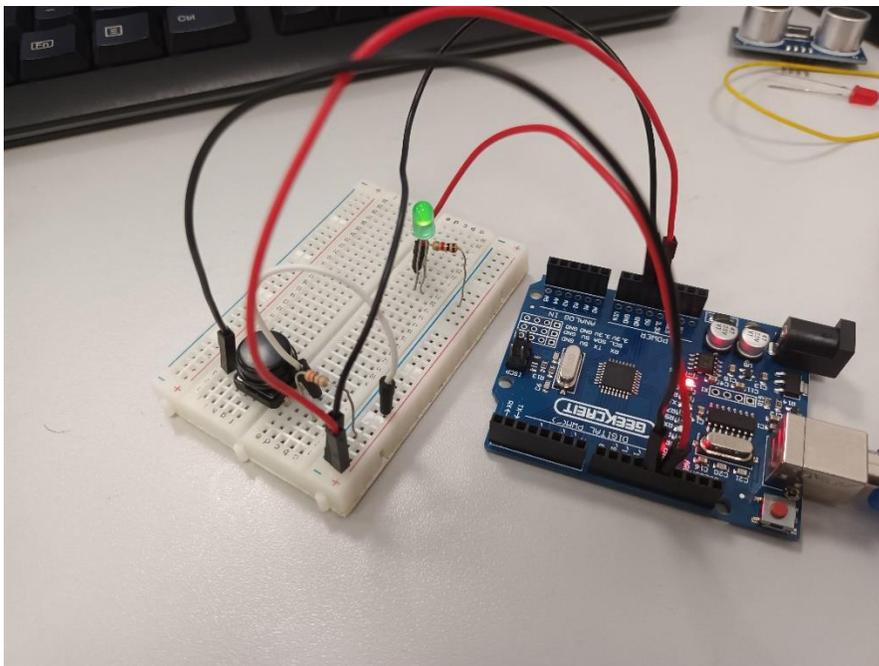
Nuno Miguel Amaral da Costa 29

Conectei uma perna do LED (o terminal mais longo, conhecido como ânodo) ao pino 8 do Arduino.

Conectei a outra perna do LED (o terminal mais curto, conhecido como cátodo) a um resistor. O valor típico para o resistor é 220 ohms, mas isso pode variar dependendo do LED e da fonte de alimentação.

E por fim conectei o outro lado do resistor (que não está conectado ao LED) ao GNND (terra) do Arduino.

Teste2



```
const byte LED = 13;
const byte BOTAO = 12;
byte valorBotao;
byte valorBotaoAnterior;
boolean ledligado = false;

void setup() {
  pinMode(LED, OUTPUT);
  pinMode(BOTAO, INPUT);
}

void load() {
  valorBotao = digitalRead(BOTAO);
  if (valorBotao != valorBotaoAnterior) {
    if (valorBotao) {
      ledligado = !ledligado;
      digitalWrite(LED, ledligado);
    }
  }
  valorBotaoAnterior = valorBotao;
  delay(10);
}
```

Este código é um exemplo de como controlar um LED usando um botão no Arduino.

Agora vou explicar o código:

‘const byte LED = 13;’ e ‘const byte BOTAO = 12;’ : Define os pinos usados para o LED (pino 13) e o botão (pino 12).

‘byte valorBotao;’ e ‘byte valorBotaoAnterior;’ : Variáveis para armazenar o estado atual e anterior do botão.

‘boolean ledligado = false;’ : Variável booleana para controlar se o LED está ligado ou desligado.

‘void setup() { pinMode(LED, OUTPUT); pinMode(BOTAO, INPUT); }’ : Configura os pinos como saída para o LED e entrada para o botão.

‘void loop() { ... }’ : O método loop() não está presente no código fornecido. Parece ser um erro de digitação, onde ‘loop()’ foi digitado como ‘load()’. O loop é onde você colocaria o código que é executado continuamente.

valorBotao = digitalRead(BOTAO); : Lê o estado do botão.

‘if (valorBotao != valorBotaoAnterior) { ... }’ : Verifica se o estado atual do botão é diferente do estado anterior.

'if (valorBotao) { ... }': Se o botão estiver pressionado, entra neste bloco de código.

'ledLigado = !ledLigado;': Inverte o estado do LED (liga se estiver desligado e desliga se estiver ligado).

'digitalWrite(LED, ledLigado);': Define o estado do LED de acordo com a variável 'ledLigado'

'valorBotaoAnterior = valorBotao;': Atualiza o estado anterior do botão para o estado atual.

'delay(10);': Pequeno atraso para evitar a leitura errática do botão devido ao bounce (efeito de salto).

Conexão dos cabos:

Conectei o terminal positivo do LED (o terminal mais longo, conhecido como ânodo) ao pino 13 do Arduino.

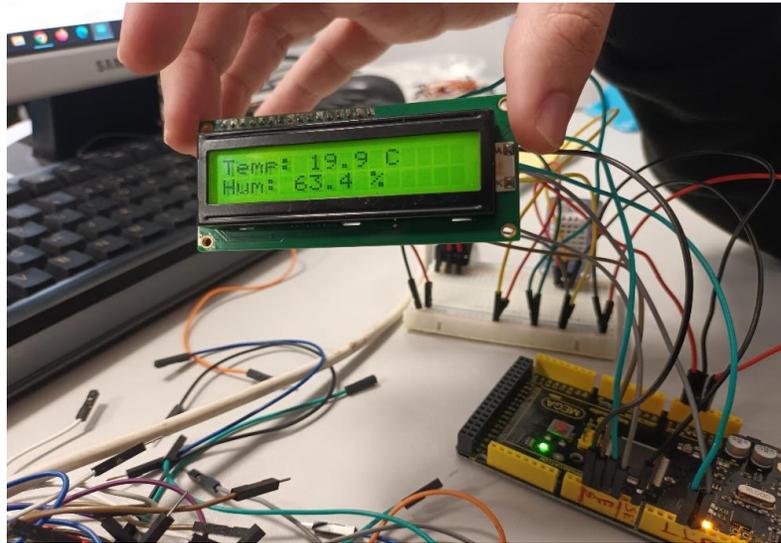
E o terminal negativo do LED (o terminal mais curto, conhecido como cátodo) a um resistor de 220 ohms.

O outro terminal do resistor está ligado a um terminal do botão, o outro terminal do botão esta ligado ao pino 12 do Arduino.

E por último conectar o resistor de pull-down do pino 12 para o GND (terra) do Arduino.

2.2 Arduinos e códigos Finais: Arduino Do Lcd 16x2 e sensor de Humidade

32 Nuno Miguel Amaral da Costa



```
#include <DHT.h>
#include <LiquidCrystal.h>

#define DHT_PIN 7 // Pino digital para o sensor DHT22
#define DHT_TYPE DHT22 // Tipo de sensor DHT (pode ser DHT11 ou DHT22)

DHT dht(DHT_PIN, DHT_TYPE); // Criação do objeto DHT

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //Pinos LCD

void setup() {
  lcd.begin(16, 2); // Inicia o LCD 16x02 (colunas, filas)
  lcd.home(); // Coloca o cursor nas coordenadas (0, 0)
  lcd.print("Temp: Hum:"); // Escreve no LCD

  dht.begin(); // Inicia o sensor DHT
}

void loop() {
  delay(2000); // Aguarda 2 segundos entre leituras

  float temperatura = dht.readTemperature(); // Leitura da temperatura
  float humidade = dht.readHumidity(); // Leitura da Humidade

  lcd.clear(); // Limpa o conteúdo do LCD
  lcd.setCursor(0, 0); // Coloca o cursor no início da primeira linha
  lcd.print("Temp: ");
  lcd.print(temperatura, 1); //Imprime a temperatura com 1 casa decimal
  lcd.print(" C");

  lcd.setCursor(0, 1); // Coloca o cursor no início da segunda linha
  lcd.print("Hum: ");
  lcd.print(humidade, 1); // Imprime a humidade com 1 casa decimal
  lcd.print(" %");
}
```

Este código é um exemplo de como ler dados de um sensor DHT22 (um sensor de temperatura e umidade) e exibir as leituras em um display LCD 16x2 utilizando um Arduino.

Agora vou explicar o código:

'#include <DHT.h>' e '#include <LiquidCrystal.h>': Inclui as bibliotecas necessárias para o sensor DHT e para o display LCD.

`#define DHT_PIN 7` e `#define DHT_TYPE DHT22`: Define o pino digital usado para o sensor DHT22 e o tipo de sensor (DHT22 neste caso).

`DHT dht(DHT_PIN, DHT_TYPE);`: Cria um objeto 'DHT' chamado 'dht' usando o pino e o tipo de sensor especificados.

`LiquidCrystal lcd(12, 11, 5, 4, 3, 2);`: Cria um objeto 'LiquidCrystal' chamado 'lcd' com os pinos conectados ao display LCD. Os parâmetros são, respetivamente, (RS, E, D4, D5, D6, D7).

`void setup() { ... }`: Define o método 'setup()', que é chamado uma vez quando o Arduino é inicializado. Neste método:

- `lcd.begin(16, 2);`: Inicializa o display LCD com 16 colunas e 2 linhas.
- `lcd.home();`: Define o cursor para a posição inicial (0, 0) do LCD.
- `lcd.print("Temp: Hum:");`: Imprime a mensagem inicial no display LCD.
- `dht.begin();`: Inicializa o sensor DHT.

`void loop() { ... }`: Define o método 'loop()', que é executado continuamente após o 'setup()'. Neste método:

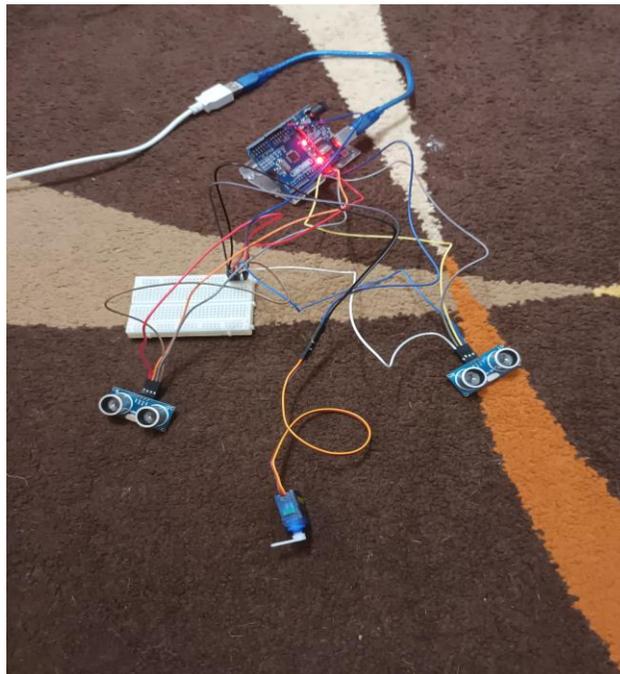
- `delay(2000);`: Aguarda 2 segundos entre as leituras.
- `float temperatura = dht.readTemperature();`: Lê a temperatura do sensor DHT.
- `float humidade = dht.readHumidity();`: Lê a umidade do sensor DHT.
- `lcd.clear();`: Limpa o conteúdo do display LCD.
- `lcd.setCursor(0, 0);`: Define o cursor para a posição inicial (0, 0) do LCD.
- `lcd.print("Temp: ");`: Imprime a mensagem "Temp: ".
- `lcd.print(temperatura, 1);`: Imprime a temperatura com uma casa decimal.
- `lcd.print(" C");`: Imprime a unidade Celsius.
- `lcd.setCursor(0, 1);`: Define o cursor para a posição (0, 1) do LCD (início da segunda linha).
- `lcd.print("Hum: ");`: Imprime a mensagem "Hum: ".
- `lcd.print(humidade, 1);`: Imprime a umidade com uma casa decimal.
- `lcd.print(" %");`: Imprime o símbolo de percentagem.

Quanto à conexão dos cabos:

34 Nuno Miguel Amaral da Costa

- *Conecte o pino de dados do sensor DHT22 (geralmente marcado como "out" ou "data") ao pino 7 do Arduino.*
- *Conecte os pinos RS, E, D4, D5, D6 e D7 do display LCD aos pinos 12, 11, 5, 4, 3 e 2 do Arduino, respectivamente.*
- *Conecte os pinos VCC e GND do sensor DHT22 e do display LCD aos pinos 5V e GND do Arduino, respectivamente.*

Arduino Sensor de Movimento e Servo Motor



```
#include <Servo.h>

int echo1 = 6; // Echo do primeiro sensor ultrassônico
int trig1 = 7; // Trigger do primeiro sensor ultrassônico
int echo2 = 9; // Echo do segundo sensor ultrassônico
int trig2 = 10; // Trigger do segundo sensor ultrassônico

float distancia1, distancia2; // Variáveis para armazenar as distâncias dos dois sensores
Servo servo1;

void setup()
{
  pinMode(echo1, INPUT);
  pinMode(trig1, OUTPUT);
  pinMode(echo2, INPUT);
  pinMode(trig2, OUTPUT);

  Serial.begin(9600);
  servo1.attach(8, 0, 180);
}

void loop()
{
  // Medição do primeiro sensor ultrassônico
  digitalWrite(trig1, LOW);
  delay(500);
  digitalWrite(trig1, HIGH);
  delay(500);
  digitalWrite(trig1, LOW);
  distancia1 = pulseIn(echo1, HIGH);
  distancia1 = distancia1 / 58;

  // Medição do segundo sensor ultrassônico
  digitalWrite(trig2, LOW);
  delay(500);
  digitalWrite(trig2, HIGH);
  delay(500);
  digitalWrite(trig2, LOW);
```

```

distancia2 = pulseIn(echo2, HIGH);
distancia2 = distancia2 / 58;

Serial.print("Distância Sensor 1: ");
Serial.println(distancia1);
Serial.print("Distância Sensor 2: ");
Serial.println(distancia2);

// Lógica para controlar o servo baseado nas distâncias medidas
if (distancia1 > 60 && distancia2 > 60)
{
  // Se ambas as distâncias forem maiores que 60 cm, mova o servo para 180 graus
  servo1.write(180);
  delay(1000); // Aguarda 1 segundo
  // Adiciona um delay adicional de 15 segundos
  delay(15000);
}
else
{
  // Caso contrário, mova o servo para 0 graus
  servo1.write(0);
}
}

```

Inclusão de bibliotecas: No início do código, a biblioteca Servo.h é incluída. Esta biblioteca é necessária para controlar o servo motor.

Declaração de pinos: São definidas quatro variáveis para representar os pinos conectados aos componentes:

echo1 e trig1 são os pinos do primeiro sensor ultrassônico.

echo2 e trig2 são os pinos do segundo sensor ultrassônico.

Variáveis de distância: distancia1 e distancia2 são variáveis para armazenar as distâncias medidas pelos sensores ultrassônicos.

Configuração no setup(): No método setup(), os pinos são configurados como entrada ou saída:

Os pinos echo1, trig1, echo2 e trig2 são configurados como saída, pois são responsáveis por enviar sinais para os sensores e receber os sinais de eco.

Nuno Miguel Amaral da Costa 37

A comunicação serial é iniciada com uma taxa de 9600 bauds para permitir a comunicação com o computador.

O servo motor é anexado ao pino 8 e configurado para ter uma faixa de rotação de 0 a 180 graus.

Loop principal: No método loop(), o código executa as seguintes ações repetidamente:

Mede a distância dos dois sensores ultrassônicos.

Exibe as distâncias medidas no monitor serial.

Com base nas distâncias medidas, controla o servo motor para mover-se para 180 graus se ambas as distâncias forem maiores que 60 cm, caso contrário, move-se para 0 graus.

Adiciona atrasos para garantir uma leitura adequada dos sensores.

Agora, em relação à conexão dos cabos:

Os pinos echo1, trig1, echo2 e trig2 devem ser conectados aos respectivos pinos dos sensores ultrassônicos. Geralmente, os sensores têm quatro pinos: VCC (5V), GND (terra), Echo e Trig. Certifique-se de conectar os pinos corretamente.

O pino 8 deve ser conectado ao pino de controle do servo motor. Geralmente, o pino de controle é o fio de sinal, que é conectado ao pino de controle do servo.

Certifique-se de conectar os pinos VCC (5V) e GND (terra) dos sensores ultrassônicos e do servo motor aos respectivos pinos 5V e GND da placa Arduino.

2.3 Maquete da Casa e Garagem com o Arduino dos sensores de movimento e servo motor incluído.



2.4 Maquete da Casa e Garagem Projeto Final Com o Arduino dos sensores de movimento e servo motor incluído e o Arduino Do Lcd 16x2 e sensor de Humidade incluído.

Nuno Miguel Amaral da Costa 39



Conclusão

40 Nuno Miguel Amaral da Costa

Com este trabalho fiquei a saber mais de Arduino, melhorei os meus conhecimentos na área, aprimorei as minhas maneiras de resolver os problemas que apareceram durante a realização do projeto, e apliquei alguns dos conhecimentos que adquiri durante 3 anos aqui na EPB.

Bibliografia

<https://www.arduino.cc/>

<https://pt.wikipedia.org/wiki/Arduino>

<https://www.tinkercad.com/dashboard>

Programação de Sistemas Informáticos
Redes de comunicação
Sistemas Operativos
Experiência profissional

COMPETÊNCIAS PESSOAIS

Portuguesa

Língua materna

Outras línguas:

	COMPREENDER		FALAR		ESCREVER
	Compreensão oral	Leitura	Interação oral	Produção oral	
Inglês A1					
	A1	A1	A1	A1	A1

© União Europeia, 2002-2018 | europa.eu/education/european-competence-portal Página 1 / 2

 Curriculum Vitae Nuno Costa

Competências digitais

AUTOAVALIAÇÃO

Processamento de informação	Comunicação	Criação de conteúdos	Segurança	Resolução de problemas

Utilizador avançado	Utilizador independente	Utilizador independente	Utilizador avançado	Utilizador independente

Níveis: utilizador básico - utilizador independente - utilizador avançado
[Competências digitais - Grelha de auto-avaliação](#)

Indique o(s) certificado(s) TIC

Descreva outras competências informáticas. Indique o contexto em que foram adquiridas. Exemplos:

- Conhecimentos Windows Desktop, Server e família Linux [Sistemas Operativos]
- Python, C#, Html, JavaScript e PHP [Linguagem de Programação]
- Access, Mysql e SQL [Base de Dados]