

Agrupamento de Escolas de Sampaio
Equipa: Sam4Tub
Projeto FLAD - Prémio Atlântico Júnior

Dossier técnico

Sam4TUB - Veículo movido a energia solar monitorizador da qualidade da água na Lagoa de Albufeira (Sesimbra)



Projeto desenvolvido no âmbito da participação do concurso FLAD - Prémio Atlântico. Durante o ano letivo 2023/24

**Sampaio
2024**

Índice

Introdução.....	3
Objetivo científico do projeto.....	4
Equipa.....	5
Descrição do protótipo.....	6
Cronograma.....	7
Fases de Execução.....	9
Materiais/ Recursos.....	11
Diagrama de Componentes.....	11
I - Estação de Controlo.....	11
Arduino Uno R3 - SMD CH340.....	11
Joystick Biaxial 10K.....	12
LoRa Radio 433MHz - Grove.....	12
II- Master.....	13
Arduino Nano 3.0.....	13
LoRa Radio 433MHz - Grove.....	13
Sensor inercial 6 eixos - I2C BMI160 - Gravity.....	14
Módulo de controlo de motores L293D.....	14
Micro motor N50 - 31k rpm.....	15
Servo MG996R 180°.....	15
III- Alimentação.....	16
Painel solar 6V 5W.....	16
Módulo MPPT alimentação Solar 6-24Vdc.....	16
LDR 5528 - VT90N2.....	17
Pilha LI-ION 18650 3.6V 3450MAh - Samsung.....	18
Suportes bateria 18650.....	18
IV- Slave.....	19
Arduino Uno R3 - SMD CH340.....	19
Módulo cartões MicroSD.....	19
RTC DS1307 e I2C EEPROM.....	19
Módulo cartões MicroSD.....	20
Pilha CR2032 3V.....	21
Sensores.....	22
Medidor pH Analógico.....	22
Sensor turbidade analógico - Gravity.....	22
Sensor analógico TDS - Gravity.....	23
Sensor analógico ORP.....	24
Sensor de temperatura água - DS18B20.....	25
Sensor de temperatura e humidade - DTH22 - Keyestudio.....	26
Sensor de radiação ultravioleta UV- UVM-30A.....	29
Parcerias.....	31

Tabela de custos	32
Conclusões.....	33
Anexos.....	34

Índice de Anexos

Anexo 1	Programação (códigos)
Anexo 2	Medição de pH com sensor de membrana de vidro e placa Arduino
Anexo 3.....	Sonar (sugestão de melhoria para o futuro)

Introdução

O presente documento apresenta o desenvolvimento do projeto de construção de um protótipo de um veículo movido a energia solar, capaz de obter dados sobre a qualidade da água e do ar em meios aquáticos, mais especificamente em lagoas.

A planificação e execução do protótipo desenrolou-se maioritariamente em sessões que decorreram no Clube de Robótica do Agrupamento de Escolas de Sampaio em Sesimbra, sendo que neste processo estiveram envolvidos vários docentes e alunos do secundário, sendo que o protótipo foi testado com amostras retiradas da Lagoa de Albufeira no concelho de Sesimbra, e numa fase final na própria Lagoa.

Neste documento são apresentados, o objetivo científico do projeto, a equipa que o desenvolveu e a descrição do protótipo/ produto final.

É também apresentado um cronograma das atividades e algumas observações de como este foi respeitado, embora com algumas alterações, decorrentes de constrangimentos que surgiram durante a sua implementação.

Para uma melhor clarificação e divulgação do modo como o protótipo é constituído, são ainda apresentadas as listas e características dos componentes e materiais utilizados bem como as linhas de código que estiveram na base da programação (anexo 1).

Em anexo apresenta-se ainda um artigo referente à calibração do sensor de pH que é utilizado no protótipo. (anexo 2)

Também em anexo é incluída a atividade de construção de um sonar, com alunos do oitavo ano, o qual será incorporado no protótipo como forma de melhoria.(anexo 3)

Objetivo científico do projeto

Criação de um barco do tipo Catamarã movido por energia solar, para análise do nível de degradação da qualidade da água na Lagoa de Albufeira, durante o período em que a lagoa se encontra isolada do Oceano Atlântico.

O plano de água da Lagoa de Albufeira está atualmente sujeito a um conjunto de usos diversificados, apresentando relevante potencial para a prática de atividades balneares e da náutica de recreio, assim como atividades aquícolas.

Considerando o dinamismo natural do sistema costeiro em que se insere, a Lagoa de Albufeira tende a fechar a ligação ao mar devido ao assoreamento, sendo necessário proceder-se à abertura artificial de uma barra de maré com o objetivo de evitar-se a eutrofização do hidrossoma lagunar. A questão da abertura da lagoa ao mar é crítica para a renovação da qualidade da água e impede a sua eutrofização, assegurando, assim, a viabilização das atividades humanas e económicas da Lagoa que dependem da qualidade deste sistema natural.

A eutrofização consiste no aumento dos nutrientes, como nitrogénio e fósforo, num ecossistema aquático, sendo responsável pelo aumento da produtividade e, conseqüentemente, a alterações em todo esse ecossistema. O aumento da produtividade ocorre, principalmente, pela proliferação excessiva de algas. Essas algas podem produzir toxinas, contaminando a água, os organismos que vivem ali e os que deles se alimentam, inclusive o Homem. No entanto, mesmo que não produza toxinas, essa proliferação excessiva de algas pode causar muitos problemas pois as algas formam uma densa camada verde sobre a superfície da água, impedindo a passagem da luminosidade para a realização de fotossíntese por parte de algas e outras que estão abaixo delas, causando-lhes a morte. A morte desses organismos leva a um aumento de matéria orgânica e de bactérias decompositoras, que utilizam o oxigénio no processo de decomposição. Isso provoca, então, a diminuição da concentração de oxigénio na água e ocasiona a morte de diversos outros organismos, como peixes. Como tal, o processo de decomposição da matéria orgânica passa a ser anaeróbica devido à falta de oxigénio e, leva à produção de gases tóxicos nesses ambientes.

No ambiente marinho a atividade de luz ultravioleta diminui com a profundidade. Esta atividade pode prejudicar certas espécies que se encontram na base da cadeia alimentar, o que leva à perturbação do ecossistema inteiro. Uma das espécies afetadas é a *Phytoplankton*, que pode sofrer danos fatais no seu organismo levando à morte. Como a *Phytoplankton* se encontra na base da cadeia alimentar, se existir uma diminuição desta espécie todas as outras terão menos alimento, levando à má nutrição ou até mesmo à morte por fome. A *Phytoplankton* absorve gases de efeito de estufa, no entanto, quando exposta a uma certa quantidade de luz ultravioleta, esta pode emitir os gases previamente absorvidos. As cianobactérias também têm um papel essencial na cadeia alimentar e, quando expostas a luz ultravioleta, podem sofrer alterações no crescimento e na sua sobrevivência, bem como a integridade da membrana celular. Os peixes podem ser afetados especialmente a nível genético, causando a fragmentação dos cromossomas entre outros danos no material genético. Como estes danos são irreversíveis, podem ser transmitidos para as futuras gerações da espécie.

Esta informação é pertinente tendo em conta que a albufeira começa a ter visitantes assim que as temperaturas começam a convidar, sendo o local privilegiado para o

divertimento de crianças, tendo em conta a ausência de correntes e a baixa profundidade. É, também, de referir que mesmo durante o inverno, as águas da Lagoa são exploradas pelo clube de vela, onde existe uma escola de Kite Surf, frequentada por adultos e crianças.

Indo além das implicações na saúde dos visitantes desta praia, é importante também referir, que os canais desta albufeira são utilizados para produção de bivalves, que serão para posterior consumo humano, e ainda que o complexo ramificado que une a Lagoa Grande à Lagoa Pequena, serve de refúgio a inúmeras espécies de aves autóctones, sendo que deram origem a um refúgio protegido com o propósito da sua observação e conservação.

Equipa

A equipa que desenvolveu o projeto é constituída por 4 alunos que foram apoiados por um conjunto de professores das diferentes áreas envolvidas.

A referir os alunos Constança Reis, Eva Rodrigues e Vicente Rasteiro do 12.º ano do Curso de Ciências e Tecnologia e Rodrigo Gralheira - 12.º ano (curso profissional de informática)

Estes alunos foram apoiados pelos professores Ana Chambel, coordenadora do clube Ciência Viva do agrupamento que apoiou na articulação com os parceiros e na componente relacionada com a divulgação, pelo professor Luís Araújo, que apoiou principalmente na área da impressão 3D, pelo professor Alberto Palma que apoiou na área da programação em arduino, pela professora Mafalda Gonçalves, professora de física que apoiou nas questões relacionadas com a área que leciona e pela professora Filipa Lopes que apoiou na seleção dos sensores a utilizar. Para além destes elementos é importante referir ainda que alguns dos professores dos alunos foram esclarecendo algumas dúvidas pontuais relacionadas com aspectos como a forma das quilhas, a flutuação ou a potência do motor.

Este foi um projeto que promoveu em grande forma a articulação curricular e a mobilização da comunidade educativa.

Descrição do protótipo

Veículo aquático construído e moldado em placas de isolamento de poliestireno extrudido XPS, o qual possui a capacidade de alojar um conjunto de sensores necessários para a realização das medições pretendidas.

A locomoção do veículo é feita por uma hélice propulsora, que sofreu algum estudo e teste para se obter a melhor eficácia com o menor consumo de energia. Acabámos por optar por uma hélice de 3 lâminas. A direção estará a cargo de 3 lemes que obedecem de forma síncrona a um servo.

A nível da alimentação, optámos pela colocação de um painel solar acoplado a um sistema de rastreamento de luz, para que se possa obter o melhor rendimento do painel em qualquer instante e pela colocação de dois painéis solares fixos, um no topo de cada quilha.

Este rastreamento é controlado de forma autónoma recorrendo a LDR's e uma placa Arduino.

Foi decidido recorrer a radiofrequência para controle remoto do barco. Isso exigiu a criação de um comando também controlado por uma placa Arduino.

Para avaliação da qualidade da água foram selecionadas as variáveis que se pretendem estudar, nomeadamente: Concentração de O₂ na água, pH, Temperatura da água e do ambiente, Turbidez (opacidade), sólidos dissolvidos na água, oxidação-redução, exposição à luz ultravioleta. Todos estes dados serão recolhidos por uma placa Arduino, associados a uma coordenada GPS, que será registada num cartão de memória para posterior recolha e partilha com as instituições que controlam o ambiente de Sesimbra.

Para avaliação da qualidade da água e do ar selecionamos algumas das variáveis a estudar, nomeadamente:

Água

- pH,
- temperatura
- total de sólidos dissolvidos (TDS),
- potencial de oxidação-redução (ORP),
- luz dispersa por sólidos suspensos (turbidity).

Ar

- temperatura
- humidade
- intensidade da radiação ultravioleta

Todos estes dados são recolhidos pelo Arduino, e registados num cartão de memória para posterior recolha e partilha com as instituições que controlam o ambiente de Sesimbra.

Cronograma

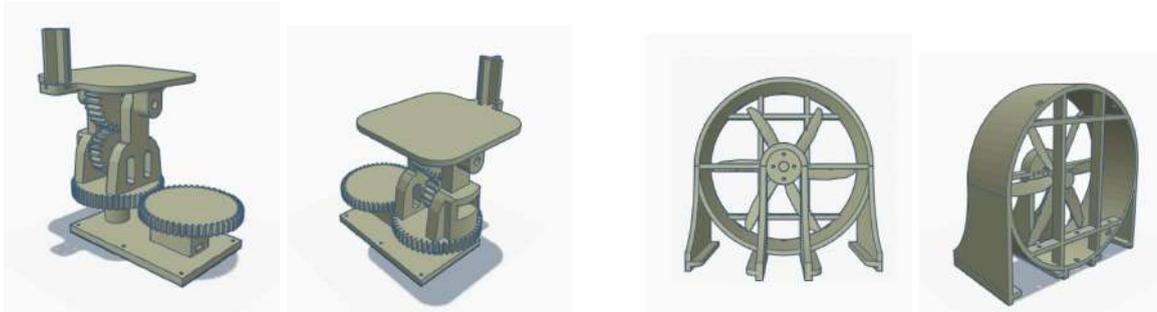
O projeto, obedeceu ao seguinte cronograma:

	setembro	outubro	novembro	dezembro	janeiro	fevereiro	março	abril	maio	junho
Planificação do projeto/ convite aos parceiros	■									
Reunião de equipa Professores/alunos	■		■	■			■		■	■
Aquisição de materiais	■	■	■							
Impressão 3D					■	■	■	■		
Montagem de circuito						■	■	■		
Programação arduíno							■	■		
Reunião Escola S. de Tecnologia do IPS								■		
Montagem de protótipo								■		
Testes de recolha de amostra de água/análise de dados									■	
Apresentação de trabalhos										■

Fig. 1 - Cronograma do projeto

Fases de Execução

Os lemes, a hélice e as engrenagens da torre que suportam o painel solar foram construídas em PLA após modelação 3D na plataforma tinkercad, abaixo representam-se as imagens destes elementos:



Figuras.3,4,5,6 - elementos que compõem o protótipo (engrenagem suporte do painel solar e hélice - imagens da plataforma tinkercad)

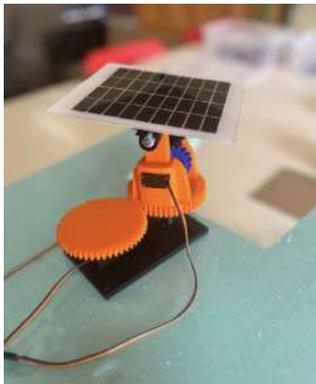


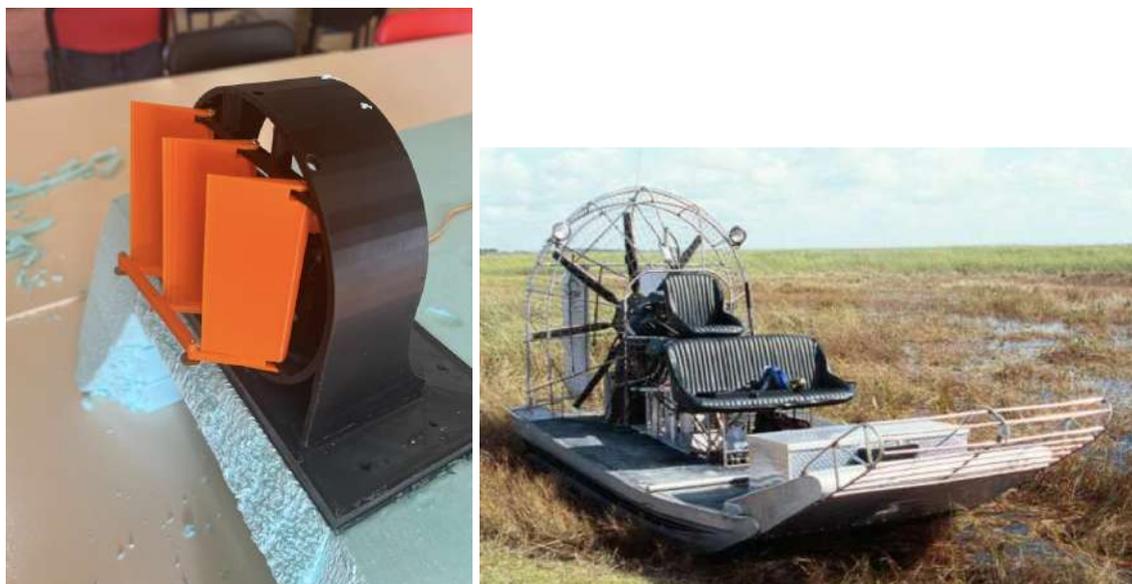
Figura 7 - Engrenagem suporte com painel solar

O veículo foi construído e moldado em placas de isolamento de Poliestireno extrudido XPS (figura 8) , não só pelas questões referidas na descrição do protótipo, mas também pela disponibilidade de equipamentos da escola (no projeto anterior foi usado PLA tendo as quilhas sido impressas nas impressoras 3D da escola o que causou vários problemas de tempo que atrasaram o desenvolvimento do projeto).



Figuras.8 - Casco do veículo construído em placas de isolamento XPS;

O veículo é movido através da impulsão de ar por uma hélice de três pás (figura 9) que se encontra inserida numa conduta formando um jato de ar que mediante a orientação de 3 lemes pode ser direccionado. Este sistema de propulsão, foi inspirado no *hovercraft* ou nos aerobarco (figura 10) (usados nos pântanos dos Everglades) e foi a opção tomada pela equipa devido a existência de algas, raízes, ramos ou detritos nos canais da Lagoa de Albufeira, que facilmente iriam bloquear uma hélice submersa.



Figuras 9 e 10 - Mecanismo de Propulsão - hélice de 3 pás e Aerobarco usado nos Everglades (Flórida)

Consideramos que neste caso, a velocidade da embarcação, não é um aspecto muito importante tendo em conta que o seu principal propósito é recolher dados com a menor perturbação do leito aquático, a fim de não alterar as medições.

Relativamente à componente de análise da qualidade da água e do ar, este processo decorreu em paralelo às restantes, foram realizadas calibrações dos diferentes sensores a par com a sua programação. Em anexo encontra-se um documento que foi construído após um desses momentos de calibragem.(ver anexo1)

Materiais/ Recursos

Apresentam-se de seguida os componentes que compõem o protótipo, nomeadamente os componentes eletrónicos e os sensores utilizados.

Diagrama de Componentes

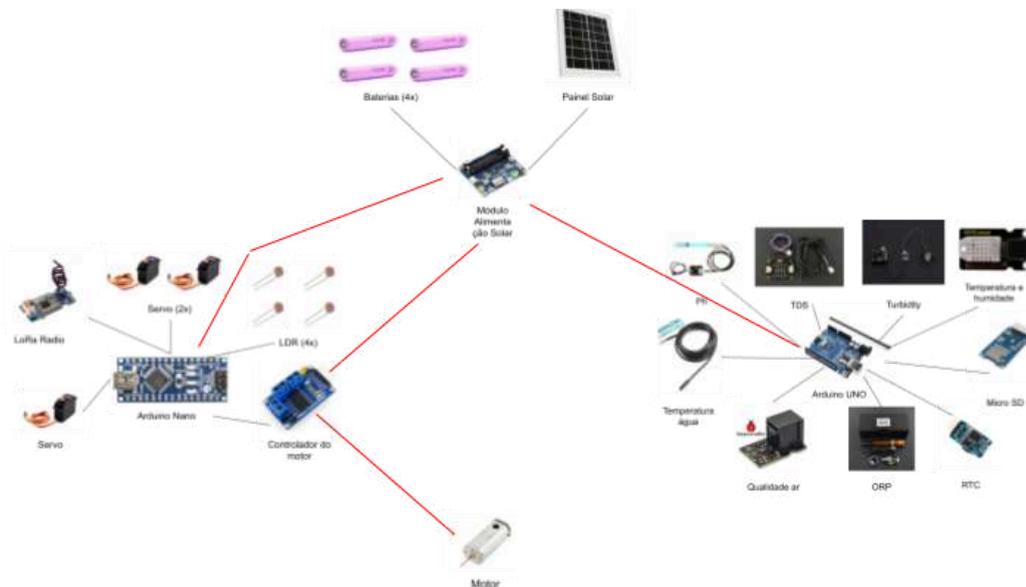
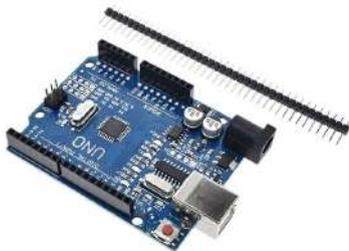


Fig 11 - Diagrama de componentes

I - Estação de Controlo

Arduino Uno R3 - SMD CH340

O Arduino Uno é uma placa de microcontrolador baseado no ATmega328. Tem 14 pinos digitais de entrada/saída (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um cristal de 16 MHz, uma conexão USB, um conector de alimentação, um ICSP e um botão reset.



Especificações:

- Microcontrolador: ATmega328P;
- Tensão Operacional: 5V;
- Tensão de Entrada: 7-12V;
- Portas Digitais: 14 (Das quais 6 Podem ser Utilizadas como PWM);
- Portas Analógicas: 6;
- Corrente Pinos I/O: 20mA;
- Corrente Pino 3.3V: 50mA;
- Memória Flash: 32KB (0,5KB Usados no Bootloader);
- SRAM: 2KB;
- EEPROM: 1KB

Fig 12 - Arduino Uno R3 - SMD CH340 • Velocidade do Clock: 16MHz.

Joystick Biaxial 10K

Este Joystick funciona através do controle de 2 potenciômetros e um botão. Duas das entradas dos potenciômetros referem-se aos eixos X e Y, sendo que o botão quando pressionado refere-se ao eixo Z. Logo este Joystick contém o total de três interfaces de entradas que são utilizadas para conectar ao eixo X, Y e Z.



Especificações:

Tensão de Operação: 3,3-5V
Marca: Keyes
Modelo: KY-023
Fácil instalação: 4 Furos
Dimensões: 3.7 x 2.5 x 3.2cm
Peso: 15g

Fig 14 - Joystick Biaxial 10K

LoRa Radio 433MHz - Grove

Módulo de comunicação LoRa® RFM95 baseado no chip SX1276.



Especificações:

- Módulo RFM95 baseado no SX1276 LoRa®
- ~ 28mA (Avg) @ + 20dBm de transmissão contínua
- ~ 8.4mA(Avg)@standby mode
- ~ 20mA (Avg) @recepção em modo, BW-500kHz
- Antena de fio simples ou conector MHF para antena externa de alto ganho
- + 20dBm - capacidade de saída de potência de 100 mW

Fig. 15 - LoRa Radio 433MHz - Grove

II- Master

Arduino Nano 3.0

O Arduino Nano é uma placa de microcontrolador baseado no ATmega328p. Tem 14 pinos digitais de entrada/saída (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um cristal de 16 MHz, uma conexão USB, um conector de alimentação, um ICSP e um botão reset.



Fig. 16 -Arduino Nano 3.0

Especificações:

- Microcontrolador: ATmega328p;
- Tensão Operacional: 5V;
- Tensão de Entrada: 7-12V;
- Portas Digitais: 14 (Das quais 6 Podem ser Utilizadas como PWM);
- Portas Analógicas: 8;
- Corrente Pinos I/O: 20mA;
- Corrente Pino 3.3V: 50mA;
- Memória Flash: 32KB (2KB Usados no Bootloader);
- SRAM: 2KB;
- EEPROM: 1KB;
- Velocidade do Clock: 16MHz.

LoRa Radio 433MHz - Grove

Módulo de comunicação LoRa® RFM95 baseado no chip SX1276.



Especificações:

- Módulo RFM95 baseado no SX1276 LoRa®
- ~ 28mA (Avg) @ + 20dBm de transmissão contínua
- ~ 8.4mA(Avg)@standby mode
- ~ 20mA (Avg) @recepção em modo, BW-500kHz
- Antena de fio simples ou conector MHF para antena externa de alto ganho
- + 20dBm - capacidade de saída de potência de 100 mW

Fig. 17 -LoRa Radio 433MHz - Grove

Sensor inercial 6 eixos - I2C BMI160 - Gravity

O sensor de movimento inercial de 6 eixos BMI160 é baseado no sensor MEMS de 6 eixos Bosch BMI160 que integra acelerômetro de 3 eixos de 16 bits com giroscópio de 3 eixos de potência ultrabaixa. Possui algoritmos inteligentes de contagem de passos integrados que podem ser lidos diretamente por meio de registros. A aceleração de 3 eixos integrada e o giroscópio de 3 eixos podem detectar corrida, condicionamento físico e outros movimentos. Chip de gerenciamento de energia LDO integrado, suporta fonte de alimentação de tensão ampla de 3,2 ~ 6V e também possui circuito de conversão de nível I2C, compatível com microcontrolador Arduino 3,3V e 5V.

Especificações



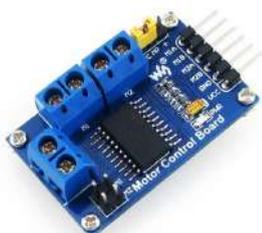
- Tensão operacional: 3,2 V ~ 6 V
- Consumo atual: <1mA
- Interface: Gravidade-IIC
- Faixa de aceleração: $\pm 2g/\pm 4g/\pm 8g/\pm 16g$
- Faixa de giroscópios:
 $\pm 125^\circ/s, \pm 250^\circ/s, \pm 500^\circ/s, \pm 1000^\circ/s, \pm 2000^\circ/s$
- Aceleração Zero-g Offset: $\pm 40mg$
- Giroscópios Zero-g Offset: $\pm 10^\circ/s$
- Frequência programável: 25/32 Hz ~ 1600 Hz
- Detecção e localização 6D
- Saída de dados de 16 bits
- Resistência ao choque: 1000gx 200us
- 2 geradores de interrupção programáveis independentes
- FIFO de 1024 bytes integrado
- Temperatura de trabalho: $-40^\circ C \sim 85^\circ C$
- Dimensão: 22X27mm/0,87x1,06 pol.

Fig. 18 - Sensor inercial 6 eixos - I2C BMI160 - Gravity

Módulo de controlo de motores L293D

Este é um shield controlador de motores de 4 canais para arduino baseado no L293D. Com este controlador pode-se usar motores DC e tensões de alimentação de 25V cm um máximo de corrente de 600mA por canal. O L293D é um controlador em ponte H.

Fig. 19 - Módulo de controlo de motores L293D



Especificações:

- Controla 4 motores DC ou 2 motores de passo ou 2 servos

- Até 4 motores DC bi-direcionais com 8 bits de seleção individual de velocidade
- Até 2 motores de passo (unipolar ou bipolar) com bobine simples, bobine dupla ou passo intercalado
- 4 Ponte-H: fornece 600mA por canal (1.2A corrente de pico) com proteção térmica, podendo controlar motores de 4.5V até 25V DC
- Resistências Pull-Down mantêm os motores desabilitados quando se liga o controlador (power-up)
- Botão de Reset
- 2 interfaces de alimentação externa para separação de alimentação do controlo e dos motores
- Compatível com Arduino Mega, Diecimila & Duemilanove

Micro motor N50 - 31k rpm



Fig. 19 - Micro motor N50 - 31k rpm

Especificações:

10 * 12 mm

Altura do motor: 25,5 MM (excluindo eixo de saída)

Eixo de saída: 1,0 MM

Comprimento do eixo de saída: 9,5 mm

Peso: 10 g

Tensão: 3,7 V

Velocidade: 27500 RPM

Corrente: 0,28 A (Corrente de Stall: 3,9A)

Tensão: 4,2 V

Velocidade: 31000 RPM

corrente:0,28 A (Corrente de parada:4,2A)

Servo MG996R 180°



Fig. 20 - Servo MG996R 180°

Especificações:

Produto: Servo MG996R

Torque: 9kg/cm(4,8V), 11kg/cm(6V)

Velocidade: 0,19s/60°(4,8V), 0,18s/60°(6V)

Ângulo de rotação: 180°

Tensão operacional: 4,8 ~ 6V

Engrenagem: metal

Banda morta: 5us

Peso: 55g

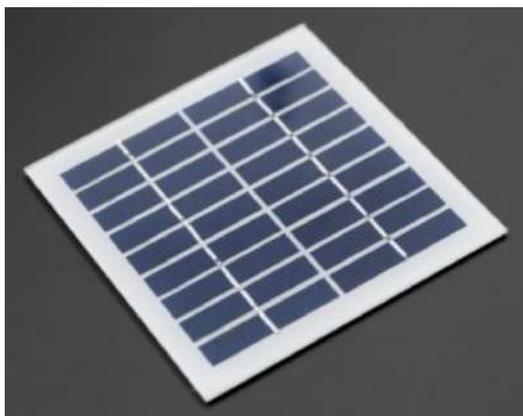
Dimensão: 40,7 mm × 19,7 mm × 42,9 mm

III- Alimentação

PAINEL SOLAR (9V 220MA)

A potência de pico é de cerca de 9V a 220mA. Também pode ser usado em série para funcionar como carregador para baterias de 12V.

O processo de laminação de vidro ultra-branco fornece uma superfície eficiente e à prova d'água para o painel. A parte traseira do painel possui fios de alumínio separados.



Especificações:

- Tensão de trabalho: 9V
- Corrente operacional: 220ma
- Tensão de circuito aberto: 9,6 V
- Corrente de curto-circuito: 500ma
- Potência: 2W
- Uso do ambiente: -40 °C ~ 80 °C
- Desempenho: corrosão, umidade
- Tamanho: comprimento 135 mm (5,31 ") x largura 125 mm (4,92")

Fig. 21 - Painel Solar

Módulo MPPT alimentação Solar 6-24Vdc

Este módulo de gerenciamento de energia solar foi projetado para painel solar de 6V ~ 24V. Ele pode carregar a bateria Li de 3,7 V por meio de painel solar ou conexão USB e fornece saída regulada de 5 V/1A.

O módulo possui função MPPT (Maximum Power Point Tracking) e circuitos de multiproteção, portanto, é capaz de continuar trabalhando com alta eficiência, estabilidade e segurança. É adequado para IoT de baixo consumo de energia solar e outros projetos de proteção ambiental.

Fig. 21 - Módulo MPPT alimentação Solar 6-24Vdc



Especificações

- Tensão de entrada do painel solar (SOLAR IN): 6V ~ 24V
- Tensão de entrada micro USB (USB IN): 5V
- Pinheader/saída USB (USB OUT): 5V 1A
- Tensão de corte de carregamento: 4,2V±1%
- Sobre a tensão de proteção de descarga: 2,9 V ± 1%
- Eficiência de carga do painel solar: ~78%

- Eficiência de carga USB:: ~82%
- Eficiência de saída de aumento da bateria de 3,7 V: ~ 86%
- Corrente quiescente máxima: <2mA
- Temperatura operacional: -40°C~85°C
- Dimensão: 65,2 mm × 56,2 mm × 22,9 mm

Servo MG996R 180°



Especificações:

Produto: Servo MG996R
 Torque: 9kg/cm(4,8V), 11kg/cm(6V)
 Velocidade: 0,19s/60°(4,8V), 0,18s/60°(6V)
 Ângulo de rotação: 180°
 Tensão operacional: 4,8 ~ 6V
 Engrenagem: metal
 Banda morta: 5us
 Peso: 55g
 Dimensão: 40,7 mm × 19,7 mm × 42,9 mm

Fig. 22 - Servo MG996R 180°

LDR 5528 - VT90N2

É um sensor de luz normal de 5 mm. Uma fotocélula muda a resistência (também chamada de fotodetector, CdS ou célula fotocondutora) dependendo da quantidade de luz à qual é exposta. Esses pequenos sensores são ótimos acionadores de luz ambiente (quando a luz da sala acender, faça alguma coisa).



Especificações:

Resistência à luz (10Lux): ~10~20K Ohm
 Resistência escura: 1M Ohm
 Tensão máxima: 150V
 Potência máxima: 100mW
 Pico Espectral: 540nm
 Modelo: 5528 (equivalente a VT90N2)
 Dimensões:
 LDR de 5 mm (1,8 x 4,5 x 5 mm)
 4mm entre pinos
 Comprimento do cabo de 30mm

Fig. 23 - LDR 5528 - VT90N2

Pilha LI-ION 18650 3.6V 3450MAh - Samsung



Especificações:

Capacidade: 3450mAh

Tensão nominal 3,6 V - 3,7 V

Tensão final do ciclo de carregamento 4,20V ± 0,05V

corrente de descarga máxima: 8A

Não descarregue mais profundo do que 2,5 V

Pólo positivo: topo plano

Química: LiNiCoMnO₂

Circuito de proteção: Sem

Diâmetro: 18,3 mm

Altura: 65 mm

Peso: 48,3g

Fig. 24 - Pilha LI-ION 18650 3.6V 3450MAh - Samsung

Suportes bateria 18650



Comprimento: 7,6 cm

Largura: 2,1 cm

Altura: 2 cm

Fig. 25 - Suporte bateria

IV- Slave

Arduino Uno R3 - SMD CH340

O Arduino Uno é uma placa de microcontrolador baseado no ATmega328. Tem 14 pinos digitais de entrada/saída (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um cristal de 16 MHz, uma conexão USB, um conector de alimentação, um ICSP e um botão reset.

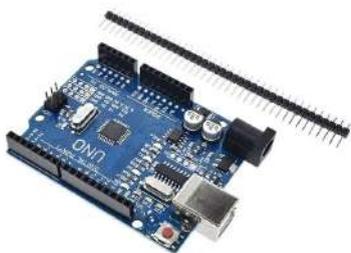


Fig. 24 -Arduino Uno R3

Especificações:

- Microcontrolador: ATmega328P;
- Tensão Operacional: 5V;
- Tensão de Entrada: 7-12V;
- Portas Digitais: 14 (Das quais 6 Podem ser Utilizadas como PWM);
- Portas Analógicas: 6;
- Corrente Pinos I/O: 20mA;
- Corrente Pino 3.3V: 50mA;
- Memória Flash: 32KB (0,5KB Usados no Bootloader);
- SRAM: 2KB;
- EEPROM: 1KB;
- Velocidade do Clock: 16MHz.

Módulo cartões MicroSD

É um módulo Micro SD(TF) da DFRobot. É compatível com cartão TF SD (comumente usado em celulares), que é o cartão mais pequeno do mercado. O módulo SD possui diversas aplicações como registrador de dados, áudio, vídeo, gráficos. Este módulo expandirá muito a capacidade que um Arduino pode fazer com sua memória limitada.

Este módulo possui interface SPI e fonte de alimentação de 5V compatível com Arduino UNO / Mega . O Pinout é totalmente compatível com o IO Expansion Shield V7 do DFRobot .



Fig. 25 -Arduino Uno R3

Especificações:

Tensão de trabalho: 5V;

Tamanho: 20x28mm(0,79x1,10")

Interface:

Compatível: MicroSD (TF)

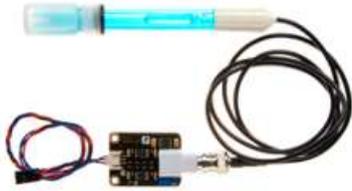
Pilha CR2032 3V



Fig. 25 -Pilha

Sensores

Medidor pH analógico



“Need to measure water quality and other parameters but haven't got any low cost pH meter? Find it difficult to use with Arduino? Here comes an analog pH meter, specially designed for Arduino controllers and has built-in simple, convenient and practical connection and features. It has an LED which works as the Power Indicator, a BNC connector and PH2.0 sensor interface. To use it, just connect the pH sensor with a BNC connector, and plug the PH2.0 interface into the analog input port of any Arduino controller. If pre-programmed, you will get the pH value easily. Comes in a compact plastic box with foams for better mobile storage.”

Fig. 26 -Medidor pH

- *DFRobot*

Como funciona:

Medição da diferença de potencial elétrico:

O sensor de pH é constituído por uma membrana de vidro sensível aos iões H^+ em solução. Quando imersa, esta membrana reage com os iões H^+ , criando uma diferença de potencial elétrico entre o interior e o exterior da membrana.

Eléctrodo de Referência:

O sensor de pH também possui um eléctrodo de referência, que tem um potencial elétrico conhecido e constante. Este eléctrodo serve para fornecer um ponto de comparação estável para a voltagem gerada pela membrana sensível aos iões H^+ .

Valor de pH:

A diferença de potencial elétrico gerada é proporcional à concentração de iões H^+ na solução. Há a conversão dessa diferença de potencial num valor de pH, utilizando a equação de Nernst, que estabelece uma relação matemática entre a diferença de potencial e a concentração de iões H^+ .

Como afeta a saúde:

Um pH mais alto, ou seja, superior a 7,5, pode provocar uma sensação de prurido no corpo e uma sensação de ardor nos olhos das pessoas. Por outro lado, um pH mais baixo, inferior a 6,5, pode causar pele seca e outros problemas de pele. Adicionalmente, tanto um pH alto como um pH baixo podem causar a morte de peixes. Tudo isto pode ser prejudicial às pessoas que frequentam a lagoa.

Especificações:

Module Power : 5.00V Module Size : 43mmx32mm

Measuring Range :0-14 pH

Measuring Temperature :0-60 °C

Accuracy : ± 0.1 pH (25 °C)

Response Time : ≤ 1 min

pH Sensor with BNC Connector
pH2.0 Interface (3 foot patch)
Gain Adjustment Potentiometer

Sensor de turbidez analógico - Gravity



“The turbidity sensor detects water quality by measuring the level of turbidity. It is able to detect suspended particles in water by measuring the light transmittance and scattering rate which changes with the amount of total suspended solids (TSS) in water. As the TSS increases, the liquid turbidity level increases. This sensor has both analog and digital signal output modes. You can select the mode according to the MCU as the threshold is adjustable in digital signal mode. Turbidity

sensors can be used in measurement of water quality in rivers and streams, wastewater and effluent measurements, sediment transport research and laboratory measurements.”

Fig. 27 -Sensor turbidez

- *DFRobot*

Como funciona:

Com este sensor é possível determinar a turbidez da água, parâmetro ligado à qualidade da água. Deteta partículas suspensas medindo a transmitância da luz e a taxa de dispersão, que mudam de acordo com a quantidade de sólidos suspensos na água.

Como afeta a saúde:

A turbidez, por si só, não tem efeitos sobre a saúde das pessoas. No entanto, pode interferir na qualidade da água, permitindo o crescimento de microrganismos nocivos, incluindo bactérias anaeróbias, parasitas e ainda podendo indicar a presença de vírus. A presença destes pode afetar a saúde de quem visita a lagoa.

Especificações:

Operating Voltage: 5V DC

Operating Current: 40mA (MAX)

Response Time:<500ms

Insulation Resistance: 100M (Min)

Output Method:Analog Analog output: 0-4.5V

Digital Output:High/Low level signal (you can adjust the threshold value by adjusting the potentiometer)

Operating Temperature: 5°C~90 °C

Storage Temperature: -10°C~90°C

Weight: 30g

Adapter Dimensions: 38mm*28mm*10mm/1.5inches *1.1inches*0.4inches

Sensor analógico TDS - Gravity

Fig. 28 -sensor TDS



TDS (Total Dissolved Solids) indicates how many milligrams of soluble solids dissolved in one liter of water. In general, the higher the TDS value, the more soluble solids dissolved in water, and the less clean the water is. Therefore, the TDS value can be used as one of the references for reflecting the cleanliness of water. A TDS pen is a widely used equipment to measure TDS value. The price is affordable, and it is easy to use, but it is not able to transmit data to the control system for online monitoring to do some water quality analysis. The professional instrument has high accuracy and can send data to the control system, but the price is expensive for the ordinary people. To this end, we have launched an analog TDS sensor kit which is compatible with Arduino, plug and play, easy to use. Matching with an Arduino controller, you can build a TDS detector easily to measure the TDS value of liquid. This product supports 3.3 ~ 5.5V wide voltage input, and 0 ~ 2.3V analog voltage output, which makes it compatible with 5V or 3.3V control system or board. The excitation source is AC signal, which can effectively prevent the probe from polarization and prolong the life of the probe, meanwhile, increase the stability of the output signal. The TDS probe is waterproof, it can be immersed in water for long time measurement.

- *DF Robot*

Como funciona:

A maioria dos metais, minerais e sais apresentam uma alta condutividade elétrica, pois são capazes de transportar cargas elétricas. Assim, quanto maior a quantidade destes sólidos na água, maior a condutividade. O medidor de TDS permite medir o nível de condutividade e convertê-lo posteriormente para um valor de sólidos totais dissolvidos, que são tipicamente indicados em ppm (partes por milhão), e necessários na determinação da qualidade da água.

Como afeta a saúde:

O alto teor de TDS pode levar, caso seja ingerida, a pedras nos rins, problemas cardíacos, diabetes, dores de estômago, insuficiência renal, insuficiência hepática e até mesmo morte em casos raros.

Características:

Wide Voltage Input: 3.3~5.5V

Good Compatibility Output: 0~2.3V analog signal output, compatible with 5V or 3.3V controller

AC Excitation Source: effectively prevent probe from polarization

Waterproof Probe

Easy to Use: Arduino compatible, simple connection, plug and play without soldering

Especificações:

Signal Transmitter Board

Input Voltage: 3.3 ~ 5.5V

Output Voltage: 0 ~ 2.3V

Working Current: 3 ~ 6mA

TDS Measurement Range: 0 ~ 1000ppm

TDS Measurement Accuracy: $\pm 10\%$ F.S. (25 °C)

Module Size: 42 * 32mm

Module Interface: PH2.0-3P

Electrode Interface: XH2.54-2P

TDS probe

Number of Needle: 2

Total Length: 83cm

Connection Interface: XH2.54-2P

Color: Black

Other: Waterproof Probe

Sensor analógico ORP (Oxidation-Reduction Potential)



Fig. 29 -Sensor ORP

In the ever-increasing need and awareness for health and safety, water has become a focus point for proper disinfection treatment as aftereffects of industrialization can become a danger for contamination of water via direct contact. Water is a critical control point of contact where proper treatment can greatly reduce these microbes, thereby reducing the chances of illness and pollution. While traditional ORP meters are hard to get or use, this Analog ORP Meter is a highly compact ORP monitoring system that fits into your Arduino. This design configuration allows the user to accurately monitor ORP without having to add any additional circuitry or components to your design. Communication with the Analog ORP Meter is done using only simple wiring and programming. You may also check the Liquid Sensor Selection Guide to get better familiar with our liquid sensor series. ORP (Oxidation-Reduction Potential) is a measure of the ability of oxidation and reduction of aqueous solution, characterization of oxidizing or reducing the relative degree. Unlike a pH measurement that follows a logarithmic curve and therefore requires more calibration adjustments, ORP follows a linear relationship and does not need instrument adjustment as much as it needs electrode maintenance. ORP has proven to be a reliable method of measuring water quality and provides the operator with a single value of measurement regardless of which product, commodity, operation, treatment or sanitizer is used, and regardless of varying field conditions or method by which chemical is applied.

- *DF Robot*

Como funciona:

Os medidores de ORP avaliam o potencial redução e oxidação de uma determinada solução. O medidor ORP é constituído por um eléctrodo (sensor) que funciona de forma semelhante ao medidor de pH. Assim como o pH de uma solução é medido com base na quantidade de iões H dissolvidos, a medição do ORP é feita com base na transferência de electrões na água. Dependendo da qualidade da água, o eléctrodo ORP atua como doador ou aceitador de electrões. Se o contaminante na água for um agente redutor, o eléctrodo ORP atua como um aceitador. Se o contaminante estiver oxidado, o eléctrodo ORP atua como doador. Durante esta transferência de electrões, uma diferença de potencial é gerada. O dispositivo portátil apresenta o valor em escala ORP através da diferença de potencial entre a solução e o ambiente exterior.

Como afeta a saúde:

Uma alteração no Potencial de Oxidação-Redução pode indicar a existência de contaminantes que podem ser prejudiciais para a nossa saúde. Para além disto, um Potencial de oxidação-redução baixo pode favorecer a proliferação de agentes patogénicos, como bactérias, e um potencial de oxidação-redução alto pode levar a proliferação de microrganismos anaeróbios, como leveduras

Aplicações:

Teste de qualidade da água

Aquicultura
Ciência Marinha
Hidropônico
Jardinagem
Química Geral

Especificações:

Module Power: +5.00V
Module Size: 40mmX27mm(1.57"x1.06")
Measuring Range: -2000mV—2000mV
Suitable Temperature: 5-70°C
Accuracy: ±10mv (25 °C)
Response Time: ≤20sec
ORP Probe with BNC Connector
PH2.0 Interface(3 foot patch)
Zero calibration button
Power Indicator LED

Sensor de temperatura água - DS18B20

Fig. 30 -Sensor de temperatura da água



Esta é uma versão à prova de água do sensor de temperatura DS18B20. Útil para quando precisa para medir algo distante, ou num ambiente molhado. Este sensor suporta até 125 °C uma vez que o cabo é revestido em PVC mas sugerimos que o mantenham abaixo de 100 °C. A qualidade do sinal do sensor não se degrada com a distância uma vez que é digital. O DS18B20 fornece leituras de temperatura 9-12 bits (configurável) através de uma interface 1-Wire, de modo que apenas um fio (e terra) precisa ser conectado a partir de uma central com microprocessador. Pode ser alimentado com 3.0~5.5V. Como cada chip DS18B20 contém um número de série único, podem ser usados vários sensores DS18B20 no mesmo barramento 1-wire. Isto permite a colocação de sensores de temperatura em vários locais. Aplicações onde este recurso é útil incluem HVAC, controlo de climatização ambiental, controlo de temperaturas no interior dos edifícios, equipamentos ou máquinas, etc.

Como funciona:

Sensor de Temperatura:

O DS18B20 possui uma junção de silício que atua como o sensor de temperatura. Esta junção é uma interface entre o material semicondutor (silício) e o ambiente ao seu redor. Quando a temperatura ao redor do sensor muda, a resistência elétrica da junção de silício também muda. Como o silício é um material semicondutor, a variação da temperatura altera a mobilidade dos elétrons no material, o que, por sua vez, afeta a resistência elétrica da junção.

Conversor:

Internamente, o DS18B20 possui um conversor analógico-digital que converte a leitura da resistência da junção de silício em um valor digital. A resolução do ADC pode ser configurada, geralmente, entre 9 e 12 bits, determinando a precisão da leitura da temperatura. A medição é depois convertida numa medição da temperatura.

Como afeta a saúde:

O aumento da temperatura altera componentes do ecossistema. Por exemplo, multiplicam-se as cianobactérias, microalgas que liberam cianotoxinas na água, levando a um processo denominado eutrofização, onde a camada de cianobactérias leva a uma diminuição da entrada de luz na água e a uma redução de oxigénio, que são prejudiciais ao plâncton, base da cadeia alimentar. As cianotoxinas podem trazer riscos à saúde pública no caso de consumo.

Especificações:

Pode ser usado com 3.0V a 5.5V alimentação/dados

$\pm 0,5^\circ\text{C}$ Precisão de -10°C a 85°C

Gama de temperaturas utilizável: -55 a 125°C (-67°F a 257°F)

9-12 bits de resolução selecionável

Usa um fio de interface - requer apenas um pino digital para comunicação

ID 64 bits exclusivo para cada sensor

Vários sensores podem compartilhar um pino

Sistema de alarme de temperatura limite

Tempo de consulta é menor que 750ms

3 interface fios:

Fio vermelho - VCC

Fio preto ou verde - GND

Fio amarelo - DATA

Tubo de aço inoxidável com diâmetro de 6 mm por 50 mm comprimento

Comprimento do cabo: 2m

SENSOR DE RADIAÇÃO ULTRAVIOLETA UV UVM-30A

Com a camada de ozono mais fina é cada vez mais importante monitorizar a radiação UV, uma vez que pode levar a queimaduras solares e outros problemas. Este sensor permite verificar a intensidade da radiação ultravioleta.



Fig. 31 Sensor de Radiação UV

Especificações:

Tensão operacional: 3-5 VCC,

Entrada de tensão: CC 0-1 VCC,

Conector de 3 pinos:

GND (-): ligação à terra,

Fora: sinal de saída do sensor,

VCC (+): alimentação do sensor,

Precisão de medição: índice +/- 1UV,

Corrente de trabalho: 0,06mA (padrão) / 0,1mA (máx.),

Comprimento de onda de resposta: 200-370 nm,

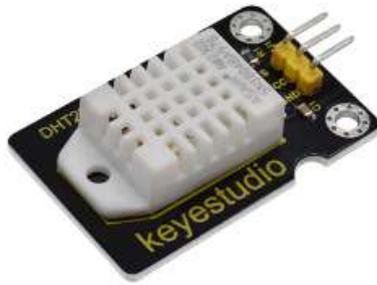
Temperatura de trabalho: -20°C - $+85^\circ\text{C}$,

Tamanho do PWB: 28x12mm

UV Index	0	UV INDEX 1	UV INDEX 2	UV INDEX 3	UV INDEX 4	UV INDEX 5
Vout(mV)	<50	227	318	408	503	606
UV Index	UV INDEX 6	UV INDEX 7	UV INDEX 8	UV INDEX 9	UV INDEX 10	UV INDEX 11+
Vout(mV)	696	795	881	976	1079	1170+

Sensor de temperatura e humidade - DHT22 - Keyestudio

Fig. 32 -sensor de temperatura e humidade (ar)



This EASY plug DHT22 digital temperature and humidity sensor is a composite Sensor which contains a calibrated digital signal output of the temperature and humidity. The dedicated digital modules collection technology and the temperature and humidity sensing technology are applied to ensure that

the product has high reliability and excellent long-term stability. Qualities of excellent quality, ultra-fast response, strong anti-interference, and high cost performance make it a wide application or even the most demanding one. The sensor comes with 2 fixed holes, very easy to mount on any other devices. For simple connection, the sensor has broken out the 4 pins into a Crystal socket, so that you can direct connect the sensor to EASY plug control board using only an RJ11 cable for communication. Applications: dehumidifier, testing and inspection equipment, consumer goods, automotive, automatic control, data loggers, weather stations, home appliances, humidity regulator, medical and other humidity measurement and control.

- *BotnRoll*

Como funciona:

Sensor de Humidade: Este sensor consiste num polímero que absorve a humidade do meio ambiente. A quantidade de humidade absorvida afeta as propriedades elétricas do polímero, especialmente a sua constante dielétrica. O sensor mede a capacitância, que varia de acordo com a humidade. Esse valor é convertido em um sinal elétrico proporcional à humidade.

Sensor de Temperatura: Um termistor é um tipo de resistência elétrica cuja resistência varia significativamente com a temperatura. A mudança na resistência do termistor é medida e convertida em um sinal elétrico proporcional à temperatura.

Especificações:

Operating voltage: DC 3.3V-5V

Interface: EASY plug RJ11

Output signal: single-bus digital signal

Humidity measurement range: 0---100%RH

Humidity measurement accuracy: $\pm 2\%$ RH

Temperature measurement range: - 40°C to 80°C

Temperature measuring accuracy: $\pm 0.5^\circ\text{C}$

Dimensions: 43mm * 26mm * 18mm

Weight: 9.3g

Parcerias

Para a execução deste projeto podemos contar com o apoio dos seguintes parceiros:

Instituto Politécnico de Setúbal - desenvolvimento do projeto e aconselhamento técnico e científico.

FABLAB Lisboa - desenvolvimento do projeto e aconselhamento técnico.

BotnRoll - aconselhamento técnico e fornecimento de material

Câmara Municipal de Sesimbra- apoio no desenvolvimento do projeto e transporte de alunos.

Tabela de custos:

Aplicação	Descrição	Preço
Controlo	Rádio RF + receptor	80,00
Alimentação	Painel + bateria	55,00
Sensores	multi sensores	80,00
Propulsão	Motor Brushless	25,00
Controle	Arduino Uno	80,00
Controle	Servos	50,00
Impressão	Bobine de filamento	50,00
Total		420,00

Conclusões

Após o desenvolvimento deste projeto a equipa pôde chegar a algumas conclusões, quer ao nível do processo científico que lhe esteve subjacente, quer ao nível das aprendizagens realizadas.

Podemos concluir que a construção de um veículo aquático engloba muitas variáveis nomeadamente variáveis relacionadas com a impulsão, a forma das quilhas, o centro de gravidade e o centro de massa, por exemplo.

Ao longo do processo foi também importante procurar informação e aconselhamento junto de diversos parceiros, quer ao nível da electrónica, da mecânica ou da fabricação/modelação em 3D, assim sendo, foi útil que os alunos se tivessem dividido em grupos e que a montagem dos diferentes componentes pudesse ser realizada em separado e apenas ligada no final da montagem do protótipo.

Para além do trabalho realizado pela equipa o projeto acabou por ter repercussões na escola, tendo gerado interesse em alunos de outros anos de escolaridade, referimos o caso de alunos de oitavo ano que se juntaram ao clube de robótica, projetaram e construíram um sonar que poderá ser utilizado no veículo (figura 33). (anexo 3)

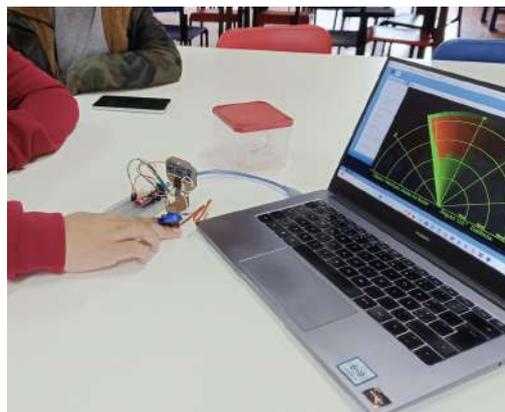


Fig. 26 -Sonar Construído por alunos de 8.º ano

Desta forma, concluímos que, o processo que envolveu todo o desenvolvimento deste projeto passando pela sua planificação até à construção e testagem do protótipo na Lagoa de Albufeira, foi gerador de muitas aprendizagens significativas, quer ao nível do método científico e de conceitos relacionados com a física, biologia e informática, mas também ao nível das competências relacionadas com o trabalho colaborativo e autónomo.

Anexos

- anexo 1Programação (códigos)
- anexo 2Medição de pH com sensor de membrana de vidro e placa Arduino
- anexo 3Sonar (sugestão de melhoria para o futuro)


```

/* -----
 * Sensors | Arduino UNO pins
 * -----
 * DHT (Air temperature and humidity) | digital 7
 * Dallas (Water temperature) | digital 4
 * pH sensor | analog A0
 * TDS sensor | analog A1
 * Turbidity | analog A2
 * ORP meter | analog A3
 * UV sensor | analog A4
 * ----- */

/* SD card attached to Arduino by SPI bus
 * -----
 * SD card | Arduino UNO pins
 * -----
 * MOSI | digital 11
 * MISO | digital 12
 * CLK | digital 13
 * CS | digital 2
 * ----- */

/* -----
 * INCLUDED LIBRARIES
 * ----- */
// for DHT sensor
#include <DHT.h>
// for One Wire communication with Dallas sensor
#include <OneWire.h>
// for DS18B20 sensor
#include <DallasTemperature.h>
// for SD card
#include <SPI.h>
#include <SD.h>

/* -----
 * GLOBAL DEFINITIONS
 * ----- */
#define DHTPIN 7 // connection of DHT sensor to Arduino pin
#define ONE_WIRE_BUS 4 // connection of Dallas sensor to Arduino pin
#define pH_PIN A0 // connection of pH sensor to Arduino pin
#define TdsSensorPin A1 // connection of TDS to Arduino pin
#define TrubidityPIN A2 // connection of Turbidity sensor to Arduino pin
#define ORPPIN A3 // connection of ORP meter to Arduino pin
#define CHIPSELECT 2 // chip select digital pin for SD card

DHT dht(DHTPIN, DHT22); // instance to communicate with DHT
OneWire oneWire(ONE_WIRE_BUS); // setup One Wire instance to communicate
DallasTemperature sensors(&oneWire); // pass the One Wire reference to Dallas sensor

File dataFile; // definition of file to store data

/* -----
 * DEBUG VARIABLE
 * ----- */
boolean debug = true; // for debugging

/* -----

```

```

* MAIN FUNCTIONS
* ----- */

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

  if (debug) {
    Serial.println("Debugging...");
    Serial.println("Air temperature (*C) | Air relative humidity (%) | Water temperature (*C) | Water pH | Water TDS (ppm/L) |
Water turbidity (NTU) | Water ORP value | UV value (mV)");
  }

  // initializing DHT sensor
  dht.begin();

  // initializing SD card
  if(!SD.begin(CHIPSELECT)){
    while(1); // don't do anything more
  }

  dataFile = SD.open("datalog.txt", FILE_WRITE);
  dataFile.println("Air temperature (*C), Air relative humidity (%), Water temperature (*C), Water pH, Water TDS (ppm/L),
Water turbidity (NTU), Water ORP value, UV voltage (mV)");
  dataFile.close();
}

void loop() {

  // wait 2 seconds before measure. For instance,
  // DHT as a time response of 2 seconds
  delay(2000);

  /* --- Read from functions --- */
  // Read air temperature and relative humidity
  float Air_Temperature, Air_RelativeHumidity;
  AirTempHumi(&Air_Temperature,&Air_RelativeHumidity);
  // Read water temperature
  float Water_Temperature = WaterTemperature();
  // Read water pH
  float Water_pH = pH();
  // Read TDS
  float Water_TDS = TDS();
  // Read Turbidity
  float Water_Turbidity = Turbidity();
  // Read ORP
  float Water_Orp = ORP();
  // Read UV (no function needed)
  int UVValue = analogRead(A4);

  if(debug){
    Serial.print(Air_Temperature,1); Serial.print(" "); // print temperature (in *C) with 1 decimal place
    Serial.print(Air_RelativeHumidity,1); Serial.print(" "); // print humidity (in %) with 1 decimal place
    Serial.print(Water_Temperature,1); Serial.print(" "); // print temperature (in *C) with 1 decimal place
    Serial.print(Water_pH,2); Serial.print(" "); // print pH with 2 decimal places
    Serial.print(Water_TDS,0); Serial.print(" "); // print TDS (in ppm) with no decimal places
    Serial.print(Water_Turbidity,0); Serial.print(" "); // print turbidity (in NTU = mg/L) with no decimal places
    Serial.print(Water_Orp,0); Serial.print(" "); // print ORP value with no decimal place
    Serial.print(UVValue*5000/1023.0); Serial.println(" "); // print UV voltage
  }

  // Opens data file
  dataFile = SD.open("datalog.txt", FILE_WRITE);

```

```

// Write data into data file
if(dataFile) {
  dataFile.print(Air_Temperature);
  dataFile.print(" , ");
  dataFile.print(Air_RelativeHumidity);
  dataFile.print(" , ");
  dataFile.print(Water_Temperature);
  dataFile.print(" , ");
  dataFile.print(Water_pH);
  dataFile.print(" , ");
  dataFile.print(Water_TDS);
  dataFile.print(" , ");
  dataFile.print(Water_Turbidity);
  dataFile.print(" , ");
  dataFile.println(Water_Orp);
  dataFile.print(" , ");
  dataFile.println(UVValue);
  dataFile.close();
}

}

/* -----
 * OTHER FUNCTIONS
 * ----- */

/* --- Get air temperature and relative humidity from DHT --- */
void AirTempHumi(float* air_temp, float* air_humi){
  *air_temp = dht.readTemperature(); // air temperature in Celsius
  *air_humi = dht.readHumidity(); // air humidity reading
}

/* --- Get water temperature from Dallas sensor --- */
float WaterTemperature(){
  // call sensors.requestTemperatures() to issue
  // a global temperature and Requests to all DS18B20 devices
  // on the One Wire bus
  sensors.requestTemperatures();
  return sensors.getTempCByIndex(0);
}

/* --- Get water pH --- */
float pH(){
  int buf[10];
  int temporary;
  // get a sample of 10 values to smooth data
  for(int i = 0; i < 10; i++){
    buf[i]=analogRead(pH_PIN);
    delay(10);
  }
  // sort data in sample
  for(int i = 0; i < 9; i++){
    for(int j = i + 1; j < 10; j++){
      if(buf[i] > buf[j]){
        temporary = buf[i];
        buf[i] = buf[j];
        buf[j] = temporary;
      }
    }
  }
}
float sum = 0;
for(int i = 2; i < 8; i++){
  sum += buf[i]; // sum the six central values from the sample
}

```

```

}
// convert the ADC value into volt
float pHValue = (float) sum * 5.0 / 1024 / 6;
// apply the calibration linear regression to convert
// the tension in volt into a pH value
pHValue = 2.06867905 * pHValue + 3.1447957132276265;
return pHValue;
}

/* --- Median filtering algorithm for TDS sensor --- */
int getMedianNum(int bArray[], int iFilterLen) {
    int bTab[iFilterLen];
    for (byte i = 0; i < iFilterLen; i++) bTab[i] = bArray[i];
    int bTemp;
    for (int j = 0; j < iFilterLen - 1; j++) {
        for (int i = 0; i < iFilterLen - j - 1; i++) {
            if (bTab[i] > bTab[i + 1]) {
                bTemp = bTab[i];
                bTab[i] = bTab[i + 1];
                bTab[i + 1] = bTemp;
            }
        }
    }
    if ((iFilterLen & 1) > 0) bTemp = bTab[(iFilterLen - 1) / 2];
    else bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2;
    return bTemp;
}

/* --- Get TDS --- */
float TDS(){
    const int SCOUNT = 30; // sum of the sample points
    float VREF = 5.0; // analog reference voltage of the Arduino ADC
    int analogBuffer[SCOUNT]; // store the analog value in the array, read from ADC
    int analogBufferTemp[SCOUNT];
    int analogBufferIndex = 0, copyIndex = 0;
    float averageVoltage = 0;
    float tdsValue = 0;
    float temperature = 25;

    static unsigned long analogSampleTimepoint = millis();
    // read the analog value from the ADC every 40 milliseconds
    if(millis() - analogSampleTimepoint > 400){
        analogSampleTimepoint = millis();
        analogBuffer[analogBufferIndex] = analogRead(TdsSensorPin); //read the analog value and store into the buffer
        analogBufferIndex++;
        if(analogBufferIndex == SCOUNT) analogBufferIndex = 0;
    }

    static unsigned long printTimepoint = millis();
    if(millis() - printTimepoint > 8000){
        printTimepoint = millis();
        for(copyIndex = 0; copyIndex < SCOUNT; copyIndex++) analogBufferTemp[copyIndex] = analogBuffer[copyIndex];
        // read the analog value more stable by the median filtering algorithm
        // and convert the voltage value
        averageVoltage = getMedianNum(analogBufferTemp, SCOUNT) * (float)VREF / 1024.0;
        // temperature compensation formula:
        // FinalResult(25°C) = FinalResult(current) / (1.0 + 0.02 * (TP - 25.0));
        float compensationCoefficient = 1.0 + 0.02 * (temperature - 25.0);
        // temperature compensation
        float compensationVoltage = averageVoltage / compensationCoefficient;
        // convert voltage value to tds value
        tdsValue = (133.42 * compensationVoltage * compensationVoltage * compensationVoltage - 255.86 * compensationVoltage *
        compensationVoltage + 857.39 * compensationVoltage) * 0.5;
    }
}

```

```

    return tdsValue;
}
}

/* --- Auxiliary function for TDS --- */
float round_to_dp(float in_value, int decimal_place){
    float multiplier = powf(10.0f, decimal_place);
    in_value = roundf(in_value * multiplier) / multiplier;
    return in_value;
}

/* --- Get Turbidity --- */
float Turbidity(){
    float voltage = 0;
    float ntu;
    // convert the analog reading (which goes from 0 to 1023) into a voltage
    // the factor 2.41 is for calibration (could be different):
    // the calibration factor is calculated based on the fact that
    // clear water will return turbidity = 4.2 NTU
    for(int i = 0; i < 800; i++) voltage += ((float)analogRead(TurbidityPIN) / 1023) * 5 * 2.41;
    voltage = voltage/800.; // average value of the 800 readings (this is to avoid noise)
    voltage = round_to_dp(voltage,2);
    if (voltage < 2.5) ntu = 3000;
    else ntu = -1120.4 * pow(voltage, 2) + 5742.3 * voltage - 4353.8;
    return ntu;
}

/* --- Auxiliary function for ORP --- */
double averagearray (int* arr, int number){
    int i;
    int maximum, minimum;
    double avg;
    long amount = 0;
    if(number <= 0){
        if(debug) Serial.println("Error: number for the array to averaging.");
        return 0;
    }
    if(number < 5){ //less than 5: calculate directly statistics
        for(i = 0; i < number; i++) amount += arr[i];
        avg = amount/number;
        return avg;
    } else {
        if(arr[0] < arr[1]){
            minimum = arr[0];
            maximum = arr[1];
        } else {
            minimum = arr[1];
            maximum = arr[0];
        }
        for(i = 2; i < number; i++){
            if(arr[i] < minimum){
                amount += minimum; // arr < minimum
                minimum = arr[i];
            } else {
                if(arr[i] > maximum){
                    amount += maximum; // arr > max
                    maximum = arr[i];
                } else {
                    amount += arr[i]; // minimum <= arr <= maximum
                }
            }
        }
        avg = (double)amount / (number - 2);
    }
}

```

```

}
return avg;
}

/* --- Get ORP value --- */
float ORP(){
float voltage = 5.0; // system voltage
float OffSet = 0.; // zero drift voltage
double orpValue;
int ArrayLength = 40;
int orpArray[ArrayLength];
int orpArrayIndex = 0;

static unsigned long orpTimer = millis(); // analog sampling interval
static unsigned long printTime = millis();

if(millis() >= orpTimer){
orpTimer = millis() + 20;
// read an analog value every 20 ms
orpArray[orpArrayIndex++] = analogRead(ORPPIN);
if(orpArrayIndex == ArrayLength) orpArrayIndex = 0;
}
// convert the analog value to ORP
orpValue = ((30 * (double)voltage * 1000) - (75 * averagearray(orpArray, ArrayLength) * voltage * 1000 / 1024)) / 75 -
Offset;
return orpValue;
}

```



Medição de pH com sensor de membrana de vidro e placa Arduino

Alberto Palma^{1,2} e Marília Silva^{1,2}

Abril de 2023

¹ Professores de Física e Química no Agrupamento de Escolas de Sampaio

² Clube de Robótica do Agrupamento de Escolas de Sampaio

E-mail para contacto: aesrobo@esec-sampaio.pt

RESUMO: Medição de pH com a utilização de módulo sensor de pH. O computador e a placa Arduino são usados para aquisição e processamento de dados. O recurso à plataforma Arduino visa fomentar a interação dos alunos com a tecnologia e tornar as atividades experimentais mais dinâmicas e atrativas.

Palavras-chave: Arduino; pH; calibração.

1 INTRODUÇÃO

O conceito Arduino nasceu por volta de 2005 [1]. A ideia inicial era baixar o custo dos microcontroladores e tornar estes circuitos integrados acessíveis a pessoas sem treino formal em eletrónica ou programação. Com o tempo, foram surgindo diversas placas eletrónicas conhecidas pela designação genérica Arduino. As mais populares, Arduino UNO e Arduino Nano, são placas de aquisição de dados construídas em volta de controladores AVR da Atmel, embora haja opções no mercado com outros controladores. As placas Arduino são *open source*, qualquer pessoa pode construir a sua própria placa escolhendo o controlador que mais lhe agrada.

A plataforma Arduino apresenta inúmeras vantagens em relação aos kits comerciais, para a realização de experiências com aquisição de dados. Para além do baixo custo, destacam-se também a simplicidade de uso e a versatilidade da placa. Um ambiente de desenvolvimento integrado (IDE) esconde as complexidades usualmente associadas ao desenvolvimento de programas para sistemas embebidos e converte a programação das placas Arduino numa quase experiência de *plug 'n' play*. Além disso, uma vasta comunidade de utilizadores, dispostos a ajudar em caso de necessidade, e uma grande coleção de bibliotecas e outros programas *open source* disponibilizados na Internet, convertem muitas das aplicações em combinações simples, das placas Arduino com módulos de sensores/atuadores existentes no mercado, e em adaptações pequenas de programas existentes.

À medida que a plataforma Arduino foi sendo adotada para a aquisição de dados, foram também surgindo módulos sensores de pH especificamente desenvolvidos para poderem ser usados em conjunto com estas placas. É o caso dos sensores desenvolvidos pela empresa DFROBOT. A Figura 1 mostra de que forma o módulo sensor de pH pode ser combinado com a placa Arduino UNO.

2 PLACA ARDUINO

A placa Arduino permite enviar e receber informações de praticamente qualquer outro sistema eletrônico. Desta forma, é possível construir, por exemplo, um sistema de captação de dados de sensores – como o pH –, processar e enviar esses dados para um sistema remoto ou mostra-los num ecrã.

A placa Arduino UNO tem uma tensão de funcionamento de 5 volts, 14 portas/pinos digitais de entrada/saída (numeradas de 0, 1, ..., 13), das quais 6 podem ser utilizadas como saídas PWM, e 6 entradas analógicas (designadas A0, A1, ..., A5). A corrente nas saídas digitais é de 40 mA.

A placa Arduino Nano é semelhante, mas mais pequena e com mais duas entradas analógicas (A7 e A8).

O módulo sensor de pH pode ser ligado da mesma forma, tanto à placa Arduino UNO como à placa Arduino Nano.

A Figura 1 mostra o esquema de combinação do sensor com a placa de aquisição de dados. Na figura, a saída do módulo é lida pela porta analógica A1 do Arduino, enquanto que o código no Apêndice B supõe a ligação do módulo à porta A0 da placa Arduino. É indiferente em qual das portas analógicas da placa é lido o sinal do módulo sensor de pH, desde que o código seja alterado na linha 7:

```
#define SensorPin A0 // pH sensor output to Arduino analog input A0
```

Figura 1: Esquema de combinação do módulo sensor de pH com a placa Arduino UNO. Nesta figura, a saída do módulo é lida pela porta A1 do Arduino.

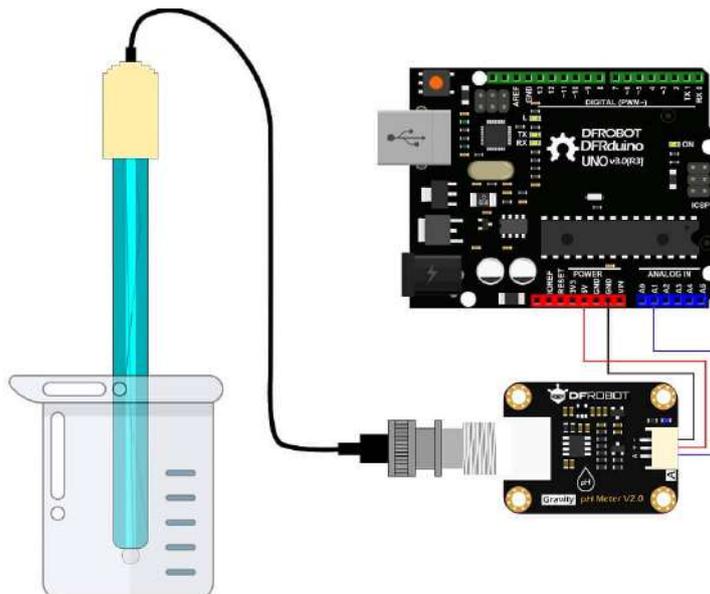


Figura reproduzida com permissão de: DFROBOT.

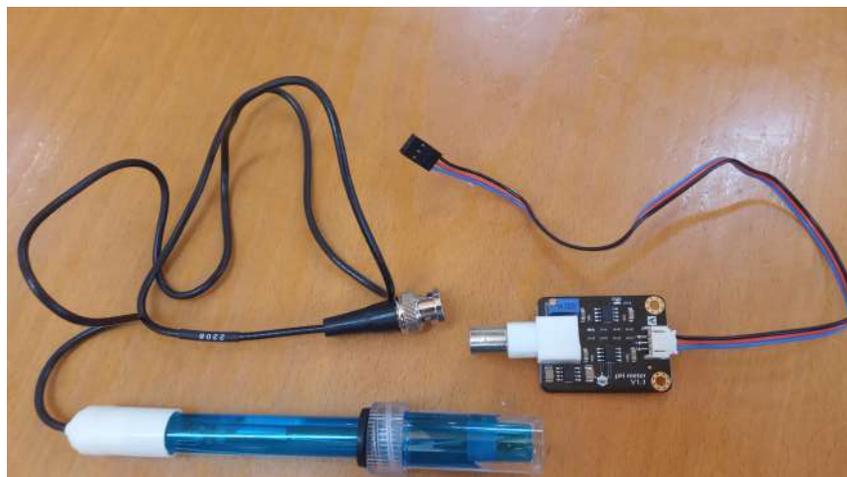
3 MÓDULO SENSOR

O módulo sensor de pH do fabricante DFROBOT (SKU* SEN0161, versão 1[†]), foi especialmente desenvolvido para ser combinado com placas Arduino, fazer medições de pH no intervalo [0,14], com uma precisão de $\pm 0,1$ a uma temperatura de 25°C, e dar uma resposta num intervalo de tempo inferior a um minuto. O módulo é composto por:

- Um sensor de pH de membrana de vidro, brevemente explicado na Secção 3.2. O sensor possui no seu interior soluções de cloreto de potássio com uma concentração de 3,3 mol/L (solução tampão interior e solução aquosa eletrolítica).
- Uma placa eletrónica para amplificar e limpar de ruído a tensão elétrica de saída do sensor. A versão 1 da placa possui uma resistência variável que permite ajustar o ganho do amplificador, com uma chave de fendas. O ajuste do ganho é importante tanto na calibração do sensor como na definição do intervalo de valores de pH acessíveis. Apesar disso, optou-se por não ajustar o ganho.

A Figura 2 é uma fotografia do módulo da DFROBOT: sensor de pH de membrana de vidro e placa eletrónica. O sensor e a placa são ligados através de uma ficha BNC.

Figura 2: Fotografia do módulo sensor de pH da DFROBOT, versão 1.



3.1 Escala de pH

O conceito de pH foi introduzido em 1909 pelo bioquímico Sören Sørensen (1868-1939) [2], diretor do departamento de química do Laboratório Carlsberg, como a forma conveniente de expressar a concentração de iões de hidrogénio numa solução. A concentração de iões de hidrogénio pode estender-se por várias ordens de grandeza, por exemplo de 12 mol/L até 10^{-14} mol/L, sendo, por isso, mais facilmente expressa na forma de um logaritmo negativo:

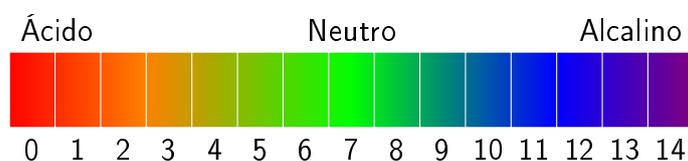
$$pH = -\log([H_3O^+]). \quad (1)$$

*O código *Stock Keeping Unit* é o número de referência único do produto, conforme aparece registado no sistema do fabricante.

[†]Estão disponíveis outras versões.

A água destilada a uma temperatura de 25°C possui uma concentração de iões $[H_3O^+] = 10^{-7}$ mol/L (e igual concentração de iões OH^-), o que corresponde a um pH de 7. A escala de pH, mostrada na Figura 3, varia entre 0 e 14 e serve para determinar o grau de acidez ou alcalinidade de soluções. As soluções neutras possuem pH 7, como a água destilada a 25°C. As soluções ácidas possuem pH inferior a 7 e as alcalinas superior a 7.

Figura 3: Escala de pH.

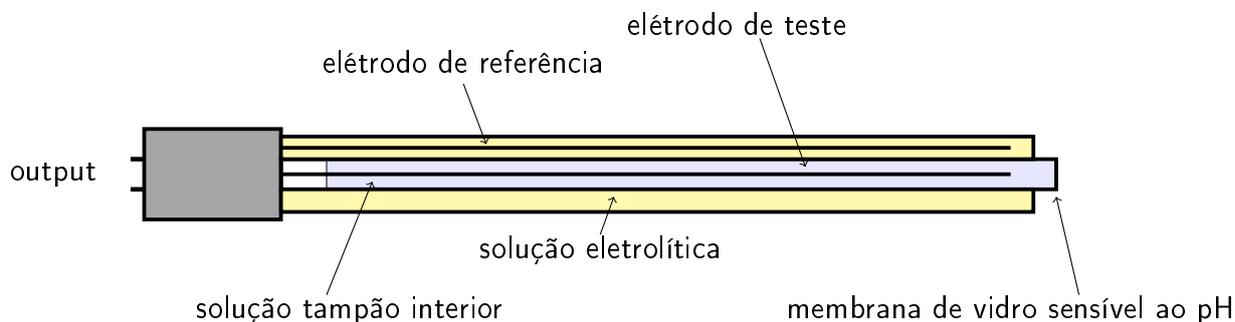


O contexto que conduziu à introdução da escala de pH foi a passagem dos antigos testes colorimétricos de indicação do grau de acidez para métodos elétricos. Nestas novas técnicas, a corrente elétrica, gerada pela migração dos iões para o eletrodo de carga elétrica oposta, era medida por um galvanómetro muito sensível. Antes da introdução da escala de pH por Sørensen não havia uma forma consensualmente aceite para expressar as concentrações de iões de hidrogénio.

3.2 Sensor de membrana de vidro

O pH pode ser determinado a partir da diferença de potencial entre dois eletrodos, medida com um voltímetro. Um dos eletrodos está imerso numa solução aquosa eletrolítica e funciona como referência (eletrodo de referência). O outro eletrodo (eletrodo de teste) está mergulhado numa solução tampão interior, separada da amostra a ser testada por uma membrana de vidro sensível ao pH, como mostra a Figura 4.

Figura 4: Representação esquemática do sensor de membrana de vidro.



Vida útil do sensor: É importante salientar que o eletrodo é um consumível: a membrana de vidro ficará menos sensível ao pH ao longo do tempo. O fabricante garante um tempo de vida útil de seis meses. Esse período depende do uso, da temperatura da amostra a ser testada, dos tipos de amostra e dos cuidados e manutenção da sonda[‡]. Em regra, é possível prolongar o tempo de vida dos eletrodos até aos dois anos. Para tal, recomenda-se:

[‡]Extremidade do sensor que entra em contacto com a amostra e onde se encontra a membrana de vidro

- A limpeza e armazenamento, sempre que terminada uma medição, para garantir uma vida útil mais longa e, também, que o eletrodo continue a ter um tempo de resposta rápido e forneça medições precisas. O sensor não deve ser guardado ligado à placa eletrônica de amplificação do sinal.
- A membrana sensível deve estar sempre hidratada, mesmo quando o eletrodo não está em utilização. Para isso, o eletrodo deve ser guardado em solução de cloreto de potássio, com concentração 3,3 mol/L, na tampa de proteção. Desta forma, o eletrodo mantém a concentração do eletrólito de referência. **Nota:** a utilização de água destilada no armazenamento irá encurtar a vida do eletrodo pois, por osmose, o eletrólito será diluído.
- Enxaguar sempre o eletrodo antes e depois das leituras (ou calibrações) com água desionizada ou destilada.

3.3 Calibração

O módulo sensor de pH foi testado, tal como foi enviado pelo vendedor (isto é, sem nenhum tipo de ajuste de ganho ou calibração), em três soluções tampão com pH 4,0, 7,0 e 10,0, respetivamente, à temperatura ambiente. O programa usado nos testes é o que se encontra no Apêndice B, com a seguinte alteração na linha 15:

```
bool calib = false; // pH sensor not calibrated if it's false
```

O sinal de saída de dados do módulo é lido por uma entrada analógica da placa Arduino. As entradas analógicas da placa Arduino estão ligadas a um conversor analógico-digital (ADC) de 10 bits e, por isso, podem assumir valores entre 0 e 1023. Isto significa que se o sinal da entrada analógica variar entre 0 e 5 volts, o monitor serial do IDE do Arduino mostrará o valor 512 quando a entrada analógica receber um sinal de 2,5 volts. A conversão entre os valores lidos pelo ADC da placa Arduino e a diferença de potencial entre os eletrodos do sensor de pH, em volts, é imediata:

$$V = ADC \times \frac{5}{1024}. \quad (2)$$

A Tabela 1 mostra os valores do ADC e as respetivas tensões, para cada uma das três soluções tampão testadas.

Tabela 1: Dados para a calibração do módulo sensor de pH.

pH solução	ADC	Tensão (V)
4,0	85	0,42
7,0	381	1,86
10,0	679	3,31

Foi ajustada uma reta aos dados, como mostra a Figura 5. O código escrito na linguagem de programação Python usado para fazer a regressão linear é apresentado no Apêndice A. A reta de ajuste tem os seguintes parâmetros:

- Coeficiente de regressão: $r^2 = 0,999996$
- Declive: $m = 2,06867905$

- Ordenada na origem: $pH_0 = 3,1447957132276265$

A reta de ajuste foi posteriormente inserida no código apresentado no Apêndice B.

3.4 Medição de pH

Após a calibração, o sensor foi usado para medir o pH de uma água mineral alcalina. A água em causa é engarrafada pela Sociedade da Água de Monchique e é publicitada com tendo um pH de 9,5. Antes e depois da leitura, o eletrodo foi enxaguado com água destilada comercial, da marca Auchan, tendo-se efetuado também medição do pH dessa água. Os resultados estão presentes na Tabela 2.

Tabela 2: Dados para a calibração do módulo sensor de pH.

Água	ADC	pH solução
destilada Auchan	300	6,17
mineral Monchique	642	9,63

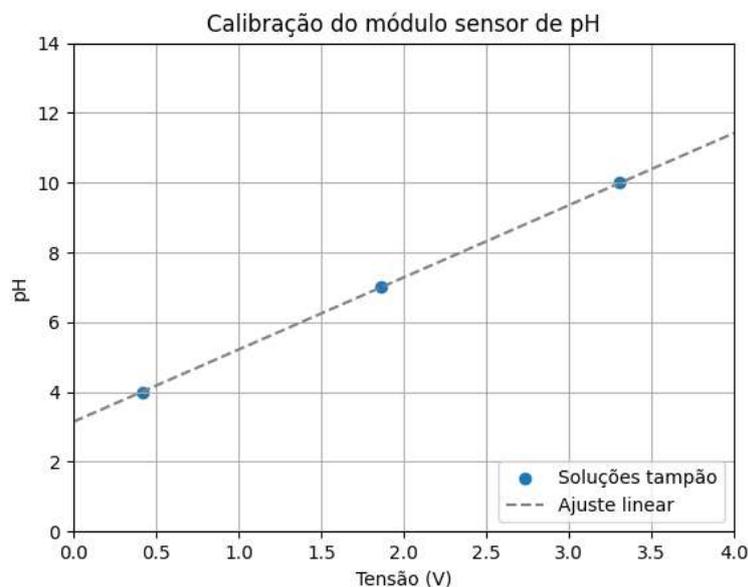
4 RESULTADOS E DISCUSSÃO

A Figura 5 mostra que a resposta do módulo é linear: a diferença de potencial entre os eletrodos e o valor do pH são diretamente proporcionais.

Após a calibração, o sensor indicou um valor de pH para a água destilada inferior a 7. O pH da água destilada purificada para fins comerciais pode variar entre 5,0 e 7,5. O valor medido está próximo do centro deste intervalo de variação do pH da água. Para a água de Monchique, foi medido um valor de pH muito próximo do valor publicitado pela empresa.

Os valores medidos estão afetados por um erro que não foi determinado. Por exemplo, o pH é

Figura 5: Ajuste linear à resposta do módulo sensor de pH.





uma grandeza influenciada pela temperatura, pelo que o facto das soluções tampão não se encontrarem a 25°C, mas sim a uma temperatura inferior que não foi medida, introduz um erro sistemático que precisa de ser estimado.

Adicionalmente, é importante notar que a reta de calibração possui uma ordenada na origem de cerca de $pH_0 \simeq 3,14$, pelo que o sensor, tal como está, permite apenas medir valores de pH no intervalo $]3,14]$, porque o Arduino não mede tensões negativas. É preciso corrigir esta situação.

5 CONCLUSÃO

O módulo sensor de pH mais simples da DFROBOT (versão 1 do SEN0161) é facilmente calibrável, rápido a responder e simples de usar. Adicionalmente, o módulo tem baixo custo.

Concluimos, assim, que o módulo sensor de pH é adequado para o uso em aulas no laboratório de Física e Química, desde que não sejam testadas amostras a temperaturas elevadas, tipicamente acima de duas vezes 25°C, ou soluções que têm ácido fluorídrico porque dissolvem a membrana de vidro.

6 REFERÊNCIAS

[1] Alisher Ismailov and Zafar Jo'rayev (2022). Study of arduino microcontroller board. *Science and Education Scietific Journal*, **3**, 3, 172-179.

[2] Sören Sörensen (1909). Über die Messung und die Bedeutung der Wasserstoffionenkoncentration bei enzymatischen Prozessen. *Biochemische Zeitschrift*, **21**, 131-304.



A Regressão Linear: código em Python

```
# import pyplot with the name plt
import matplotlib.pyplot as plt
# import numpy library with the name np
import numpy as np

# import LinearRegression class from scikit-learn package
from sklearn.linear_model import LinearRegression

# vetores
adc = [85,381,679] # ADC readings
buf = [4,7,10] # pH of buffer solutions

# convert ADC readings into volts
V = []
for i in adc:
    V.append(i * 5 / 1024)

# arrays for linear regression
x = np.array([V[0],V[1],V[2]]).reshape((-1,1))
y = np.array([buf[0],buf[1],buf[2]])

# LinearRegression instance
model = LinearRegression()

model.fit(x,y)

r_sq = model.score(x,y)
print(f"Regression coefficient:_{r_sq}")
print(f"Intercept:_{model.intercept_}")
print(f"Slope:_{model.coef_}")
```



B Medição de pH: código para Arduino UNO ou Nano

```
1 /* Code for testing the pH Sensor from DFROBOT
2 * SKU: SEN0161
3 * Arduino: UNO or Nano
4 * Date: April 2023
5 */
6
7 #define SensorPin A0 // pH sensor output to Arduino analog input A0
8 #define LEDBuiltIn 13 // Arduino LED built-in
9
10 unsigned long int avgValue; // variable to store the sensor feedback
11 int buf[10]; // array to store data (10 readings)
12 int temp; // temporary variable
13
14 bool debug = true; // No debug messages if it's false
15 bool calib = true; // pH sensor not calibrated if it's false
16
17 void setup() {
18     pinMode(LEDBuiltIn,OUTPUT); // LED built-in defined as output
19     Serial.begin(9600); // initializing serial monitor
20     Serial.println("READY!"); // test serial monitor
21 }
22
23 void loop() {
24
25     /*
26     * Get a sample of 10 values to smooth the data
27     */
28     for (int i = 0; i < 10; i++) {
29         buf[i] = analogRead(SensorPin);
30         if (debug) Serial.println (buf[i]);
31         delay(10);
32     }
33
34     if (debug) Serial.println ("SORTING!");
35
36     /*
37     * Sort an array in ascending order
38     */
39
40     for (int i = 0; i < 9; i++) {
41         for (int j = i + 1; j < 10; j++) {
42             if (buf[i] > buf[j]) {
43                 temp = buf[i];
44                 buf[i] = buf[j];
```



```
45         buf[j] = temp;
46     }
47 }
48 }
49
50 if (debug) Serial.println ("Sorted_data:_");
51
52 avgValue = 0;
53 for (int i = 2; i < 8; i++) {
54     avgValue += buf[i]; // sum the 6 central values from the data
55     sample
56     if (debug) Serial.println (buf[i]);
57 }
58
59 /*
60  * Convert the ADC value into volt
61  */
62
63 float pHValue = (float) avgValue * 5.0 / 1024 / 6;
64
65 /*
66  * Convert the volt sensor reading into pH value
67  */
68
69 if (!calib) {
70     pHValue = 3.5 * pHValue;
71 } else {
72     pHValue = 2.06867905 * pHValue + 3.1447957132276265;
73 }
74
75 /*
76  * Print the pH measurement
77  */
78 Serial.print ("_____pH:_");
79 Serial.print (pHValue,2); // result with two decimal places
80 Serial.println ("_"); // blank line
81
82 /*
83  * Use the Arduino LED built-in to signal the end of the loop
84  */
85 digitalWrite(LEDBuiltIn,HIGH);
86 delay(800);
87 digitalWrite(LEDBuiltIn,LOW);
88 }
```

CONSTRUÇÃO DE UM SONAR COM O MICROCONTROLADOR ARDUINO, O SENSOR HC-SR04 E O SOFTWARE PROCESSING

Diana Rosendo
Aluna da turma 8.ºA
Escola Básica do Castelo
Agrupamento de Escolas de Sampaio
Rua dos Casais Ricos
2970-577 Sampaio

Vanessa Rocha
Aluna da turma 8.ºA
Escola Básica do Castelo
Agrupamento de Escolas de Sampaio
Rua dos Casais Ricos
2970-577 Sampaio

Ano Letivo 2023/2024

RESUMO

Sonar é o acrónimo inglês para *SOund Navigation And Ranging*, cuja tradução para português é navegação e localização por som. O sonar teve a sua origem no início do século XX, quando um grupo de cientistas desenvolveu um sistema acústico para detetar corpos submersos. O presente artigo mostra em que medida é que a construção de um sonar pode ser realizada por alunos do Ensino Básico, recorrendo ao microcontrolador Arduino e ao sensor HC-SR04. O desenvolvimento do dispositivo aqui descrito funciona sobretudo como exemplo de um projeto capaz de cativar, entusiasmar e motivar os estudantes para a aprendizagem da Física e Química, numa ambiente que se pretende que seja caracterizado pela curiosidade e procura de conhecimento integrado.

Palavras-chave sonar · Arduino · sensor HC-SR04 · Processing · ultrassons

Os professores precisam de integrar plenamente a tecnologia no currículo, em vez de a ver apenas como um complemento, uma reflexão ou um evento.

Heidi Hays Jacobs

1 Introdução

Resguardado dos olhares sob a superfície das águas, um imenso mundo intrigante e muitas vezes inexplorado desafia a nossa perceção. O sonar, dispositivo de deteção de objetos submersos por meio da emissão e receção de ondas sonoras, tem sido essencial para desvendar os mistérios que residem sob a superfície aquática.

A história do sonar remonta ao início do século XX e foi impulsionada por dois acontecimentos dramáticos: o naufrágio do RMS Titanic e a Primeira Guerra Mundial [1]. A colisão entre o maior e mais luxuoso navio de passageiros do seu tempo e um icebergue, por um lado, e a guerra de corso¹ levada a cabo por submarinos, por outro lado, conduziram à invenção do sonar como resposta fundamental e revolucionária para o problema da deteção de corpos submersos.

Este trabalho propõe a construção de um sonar utilizando a plataforma Arduino e o sensor HC-SR04, reconhecendo a importância de tornar acessíveis tecnologias antes reservadas a laboratórios mais avançados. O sonar, neste contexto, transcende a sua aplicação convencional, revelando-se uma ferramenta versátil e de baixo custo com potencial para aplicações práticas que vão além das águas profundas.

¹Esta é uma guerra onde se privilegia a destruição ou interrupção das cadeias logísticas do inimigo.

Além disso, é essencial destacar que o estudo do sonar se inscreve no âmbito dos estudos de Física e Química no Ensino Básico. No oitavo ano, quando são abordados os princípios do som, o sonar pode ser introduzido como uma aplicação prática e fascinante, ligando os conceitos teóricos à sua utilização prática na detecção de corpos.

Adicionalmente, incorporamos a ferramenta de software Processing neste projeto, proporcionando aos estudantes uma valiosa oportunidade de aprender e aplicar conhecimentos em programação e visualização gráfica em tempo real. O uso do software Processing não só permite a representação gráfica dos dados recolhidos pelo sonar de forma intuitiva, mas também destaca a importância do desenvolvimento de competências em Ciência, Tecnologia, Engenharia e Matemática (STEM).

O presente artigo encontra-se estruturado em oito partes: a primeira corresponde a uma descrição breve de conceitos relacionados com o sonar e ultrassons; a segunda parte descreve os materiais e métodos usados no projeto; a terceira parte diz respeito ao desenvolvimento do código para Arduino e Processing; a quarta parte relata a importância da simulação para a construção do sonar; a quinta parte descreve o processo de desenho e fabricação de peças de suporte para o sonar; a sexta parte aborda os resultados obtidos e as implicações do projeto; segue-se a discussão dos resultados e sugestões de melhoria; por fim, são apresentadas as conclusões.

2 Sonar e ultrassons

O sonar é uma tecnologia que utiliza ondas sonoras para detecção, localização e identificação de corpos submersos. Esse sistema opera com base no princípio da ecolocalização (Figura 1), onde séries de ondas (também chamadas de pulsos) são emitidas a intervalos regulares e o tempo de retorno dos ecos é medido para determinar a distância até ao objeto. Além disso, o sonar pode fornecer informações sobre a forma e características do objeto detetado.

A distância entre o sonar e o objeto é calculada com base na seguinte fórmula:

$$d_{\text{objeto}} = \frac{1}{2} \times v_{\text{som}} \times \Delta t_{\text{ida e volta}}, \quad (1)$$

onde v_{som} é a velocidade do som no meio em que a onda sonora se propaga e $\Delta t_{\text{ida e volta}}$ é o intervalo de tempo medido entre o instante em que os pulsos são emitidos pela fonte e o instante em que os ecos desses pulsos são recebidos no recetor.

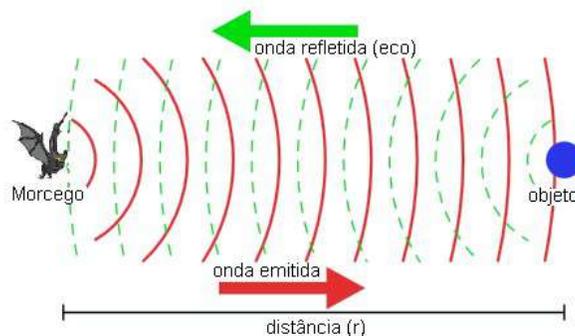


Figura 1: O morcego emite ultrassons que são refletidos pelo objeto permitindo-lhe saber a distância a que ele se encontra do objeto.

Fonte: Wikipédia.

O sonar utiliza ultrassons, o dispositivo emite rajadas curtas de ondas sonoras com frequências superiores àquelas audíveis pelo ouvido humano, geralmente acima de 20000 hertz.

O facto do ouvido humano não ser sensível a sons com frequências acima de 20 kHz torna-os valiosos em diversas aplicações. Por exemplo, os ultrassons são utilizados em medicina para construir imagens que servem de base à realização de diagnósticos (imagiologia médica). Também existem ultrassons terapêuticos, como os que são utilizados no tratamento de coágulos sanguíneos (trombólise por ultrassom), na fragmentação de cálculos renais (litotripsia por ultrassom), na destruição seletiva de células tumorais (ablação por ultrassom), na cicatrização de tecidos, redução de inflamações e alívio de dores, na introdução de medicamentos em células alvo, entre outras aplicações.

Não é apenas na medicina que podemos usar ultrassons. Na indústria alimentar, por exemplo, os ultrassons podem ser empregues na avaliação da composição química de alimentos, na descontaminação de matrizes alimentares, na extração de compostos, na emulsificação e pasteurização de alimentos, entre outras aplicações [2,3,4].

3 Materiais e métodos

O sonar que é aqui descrito tem na placa Arduino Nano a sua peça central. Esta pequena placa eletrônica funciona como a unidade de controlo responsável por gerar os pulsos de ultrassom e processar os ecos recebidos. O microcontrolador Arduino está ligado ao sensor HC-SR04, coordenando a emissão de pulsos e a medição dos tempos de retorno.

O sensor HC-SR04 é um componente crucial que utiliza a técnica do eco para medir a distância até um objeto. Ele possui dois transdutores ultrassônicos, um para emitir os pulsos e outro para receber os ecos. A ligação do sensor ao Arduino é estabelecida por meio de quatro pinos: VCC (alimentação), GND (terra), TRIG (pulso de emissão) e ECHO (pulso de retorno). A tabela 1 mostra de que forma o sensor deve ser ligado à placa Arduino.

Pinos do HC-SR04	Portas do Arduino
VCC	5 V
GND	GND (0 V)
TRIG	Saída digital
ECHO	Entrada digital

Tabela 1: Ligação física entre os pinos do sensor HC-SR04 e as portas do Arduino.

O processo de aquisição e processamento de dados segue as seguintes etapas:

1. Emissão de pulsos: a placa Arduino envia um sinal de pulso através da porta TRIG para o sensor HC-SR04, que emite uma série de ondas ultrassônicas.
2. Recepção de pulsos: o sensor recebe os pulsos refletidos por objetos no ambiente e converte-os em sinais elétricos.
3. Medição do tempo de retorno: o Arduino mede o tempo decorrido entre a emissão do pulso e a recepção do eco.
4. Cálculo da distância: com base no tempo de retorno, a distância até ao objeto é calculada usando a fórmula da velocidade do som (equação 1).
5. Visualização em tempo real: os dados adquiridos são transmitidos para o ambiente de programação Processing, onde são processados e visualizados em tempo real, oferecendo uma representação gráfica da topografia à frente do sensor.

Este método permite uma deteção precisa da distância até aos objetos presentes na frente do sensor, até onde o alcance do sensor permitir, oferecendo uma base sólida para a aplicação prática do sonar construído.

Para cobrir um ângulo de 180 graus e aumentar a abrangência da deteção, incorporamos um pequeno motor servo ao projeto. Este motor é acionado pelo Arduino e permite o movimento controlado do sensor HC-SR04 de um lado para o outro. Com isso, o sistema pode varrer uma área mais ampla, otimizando a deteção de objetos na vizinhança do sensor.

A tabela 2 mostra a lista completa dos materiais usados na construção do sonar.

Componentes
1 Arduino Nano
1 sensor ultrassônico HC-SR04
1 motor micro servo SG90 (180°)
1 placa de ensaio
Fios condutores jumpers

Tabela 2: Lista de materiais usados na construção do sonar.

4 Desenvolvimento do código

Vamos precisar de programar em dois locais: no ambiente de desenvolvimento integrado (IDE) para placas Arduino, o interface que contém as ferramentas essenciais para escrever, compilar e fazer a transferência de código para as placas Arduino, e no ambiente de desenvolvimento integrado (IDE) Processing, projetado para facilitar a criação de projetos visuais interativos, gráficos e multimídia.

4.1 Manipulação do motor servo com o Arduino

O motor micro servo SG90 possui 3 pinos: alimentação positiva (5 V), terra (GND) e pino digital de entrada/saída.

O código que o Arduino usa para controlar o motor é o seguinte:

```
#include <Servo.h>

Servo motorServo;

int servoPin = 5;
int angle = 90;

void setup() {
  motorServo.attach(servoPin);
}

void loop() {
  motorServo.write(angle);
  delay(20);
  angle = (angle + 1) % 180;
}
```

O código começa por incluir a biblioteca *Servo.h*, onde estão definidas funcionalidades específicas que permitem economizar tempo e esforço na escrita do código. Seguidamente é definida a instância com o nome *motorServo* da classe *Servo*, definida na biblioteca e que estabelece as propriedades e comportamentos associados ao motor servo. É através da instância (ou objeto) chamada *motorServo* que acedemos e manipulamos as características específicas do motor servo durante a execução do programa. A porta digital 5 da placa Arduino é definida como a porta de controlo do motor servo.

A instrução *motorServo.attach(servoPin)* associa o objeto *motorServo* à porta digital *servoPin* do Arduino. Esta instrução é executada no interior da função *setup*, que é a função que é executada apenas uma vez quando o programa corre e é comumente usada para realizar configurações iniciais.

Após a execução da função *setup*, a função *loop* é executada repetidamente num ciclo infinito. O motor é colocado inicialmente no ângulo 90° com a instrução *motorServo.write(angle)* e seguidamente o ângulo é incrementado em 1°. A instrução *delay(20)* faz uma pausa de 20 milissegundos na execução do código para que o motor possa ser posicionado no ângulo pretendido. Seguidamente, é feito o incremento de 1° no ângulo, garantindo que o ângulo é incrementado sem sair do intervalo [0°; 180°]. Assim, quando o último ângulo for 179°, o próximo ângulo será 0°.

4.2 Controlo do sensor ultrassónico com o Arduino

O sensor ultrassónico HC-SR04 é capaz de medir distâncias de 2 centímetros a 4 metros com ótima precisão. O sensor possui 4 pinos: alimentação positiva (5 V), terra (GND), pino para o controlo do envio de pulsos e pino para a receção dos sinais elétricos dos ecos.

O código utilizado para controlar o motor foi enriquecido com as seguintes linhas, para que o Arduino possa controlar também o sensor ultrassónico HC-SR04:

```

int trigPin = 9;
int echoPin = 10;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  long duration, distance;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);

  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2.)*0.0343;

  Serial.print("Angle: ");
  Serial.print(angle);
  Serial.print(", Distance: ");
  Serial.println(distance);
}

```

As portas digitais 9 e 10 da placa Arduino são usadas para a comunicação com o sensor. Estas escolhas são indicadas no início do código.

De seguida, a função `setup()` é usada para configurar e inicializar os parâmetros iniciais do programa. Por exemplo, a instrução `Serial.begin(9600)` serve para iniciar a comunicação serial entre o Arduino e outros dispositivos, neste caso o computador. Neste tipo de comunicação, os dados são transmitidos bit a bit sequencialmente, utilizando um único fio para transmitir e outro para receber. A função `begin(9600)` configura a taxa de transmissão (*baund rate*) da comunicação serial. Neste caso, a taxa de transmissão é definida como 9600 bits por segundo (bps).

A função `pinMode()` é usada para indicar se uma determinada porta da placa Arduino vai ser usada como saída (*OUTPUT*) ou como entrada (*INPUT*) de sinais elétricos.

A função `loop()` é chamada continuamente após a execução da função `setup()`. Enquanto o Arduino estiver ligado, a função `loop()` será executada repetidamente, permitindo a criação de um ciclo contínuo para o funcionamento do programa. Neste ciclo, a função `digitalWrite()` é usada para colocar tensão nas portas do Arduino. Neste caso, primeiro coloca-se uma tensão de 0 volts (*LOW*) na porta `trigPin`, seguidamente é feita uma pausa muitíssimo curta de 2 milissegundos², depois coloca-se uma tensão de 5 volts (*HIGH*) na porta `trigPin` para gerar um pulso ultrassónico. Quando a onda sonora é refletida por um objeto no campo de ação do sensor, é produzido um sinal elétrico de 5 volts na porta `echoPin`. O tempo de ida e volta do pulso ultrassónico é calculado usando a função `pulseIn()`. É a partir deste tempo que é calculada a distância, em centímetros, até ao objeto, admitindo que o som se propaga no ar a uma velocidade de 343 m/s (recordemos a equação I). Os resultados são mostrados no monitor serial, usando as funções `print()` e `println()`.

4.3 Visualização dos dados no ambiente Processing

Após a leitura da distância, a placa Arduino envia os dados para o ambiente Processing através da comunicação serial. No ambiente Processing, os dados recebidos do Arduino são processados para criar uma representação visual da distância medida. Geralmente, isto envolve desenhar uma forma que representa a distância em relação a algum ponto de referência, que neste caso é a localização do sonar.

O programa que corre na placa Arduino continua a ler e a transmitir os dados do sensor num ciclo contínuo, permitindo a atualização em tempo real da visualização no ambiente Processing.

²As instruções `delay()` e `delayMicroseconds()` são ambas usadas para impor pausas à execução do código. A diferença é que a segunda é usada quando queremos fazer pausas muitíssimo mais curtas.

5 Simulação do circuito

A simulação desempenha um papel importante, tanto na projeção dos circuitos elétricos como no seu ensaio antes da implementação real.

Ao simularmos o circuito no simulador TinkerCAD, um dos programas mais populares no seio da comunidade educativa, podemos verificar se todas as ligações estão corretas, se os componentes funcionam como esperado e se o código está bem escrito. Isso ajuda a evitar problemas durante a montagem física do circuito.

A simulação permite testar diferentes configurações e ajustes sem a necessidade de montar e desmontar o circuito repetidamente, o que economiza tempo e recursos, especialmente em projetos complexos.

Os alunos podem explorar e experimentar os diferentes componentes, como placas Arduino e sensores, antes mesmo de adquiri-los. Isso promove a aprendizagem prática e ajuda a desenvolver as capacidades relacionadas com o desenvolvimento de projetos.

Além disso, o TinkerCAD pode ser acedido em qualquer lugar com ligação à internet, permitindo que os alunos trabalhem nos seus projetos em casa, na escola ou em qualquer outro lugar.

Durante a simulação, ilustrada na figura 2, foi observado o comportamento do sonar em ação. Esta etapa permitiu a validação do projeto e a identificação de possíveis ajustes antes da implementação real.

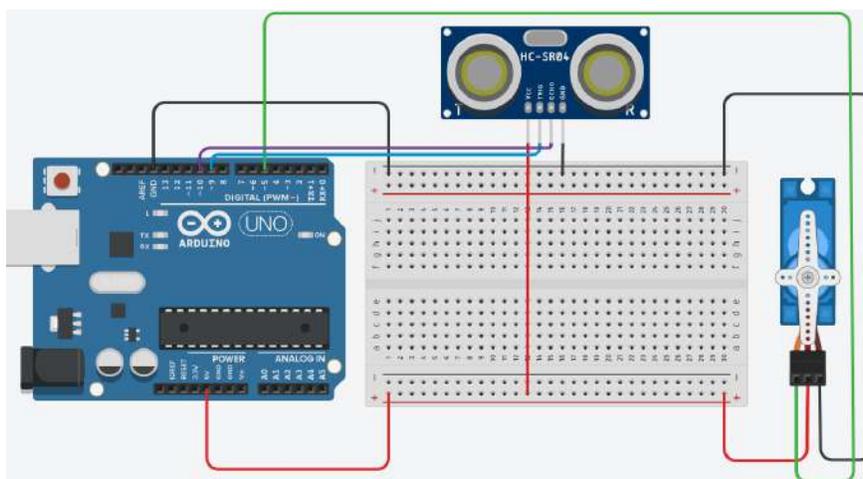


Figura 2: Circuito do sonar simulado no programa TinkerCAD.

6 Modelação 3D e impressão das peças do Sonar

A construção do sonar não se limitou apenas à programação e à montagem eletrónica, também envolveu um processo de modelagem 3D e a impressão das peças necessárias para suportar o sensor HC-SR04 e o motor micro servo. O processo de desenho 3D foi realizado no programa TinkerCAD e as peças foram impressas usando filamento PLA (ácido polilático) numa impressora 3D.

Foram redesenhadas, a partir de modelos disponibilizados no repositório Thingiverse, duas peças fundamentais para o funcionamento do sonar: uma para suporte do sensor HC-SR04 e outra para alojamento do motor micro servo e base do sonar.

O desenho das peças foi cuidadosamente ajustado para garantir uma montagem estável e segura, permitindo o posicionamento preciso dos componentes eletrónicos.

Durante o processo de desenho, foram considerados fatores como a compatibilidade com os componentes eletrónicos, resistência e facilidade de montagem. Foram incorporadas características como aberturas para a passagem de cabos e parafusos de fixação, para facilitar a montagem e manutenção do sonar.

Após a conclusão do desenho das peças, foram exportados os arquivos num formato adequado à impressão 3D. As peças foram então impressas em PLA para garantir durabilidade e resistência.

Uma vez concluída a impressão das peças e eliminadas algumas imperfeições com uma lima, estas foram montadas integrando-se perfeitamente com os demais componentes do sonar.

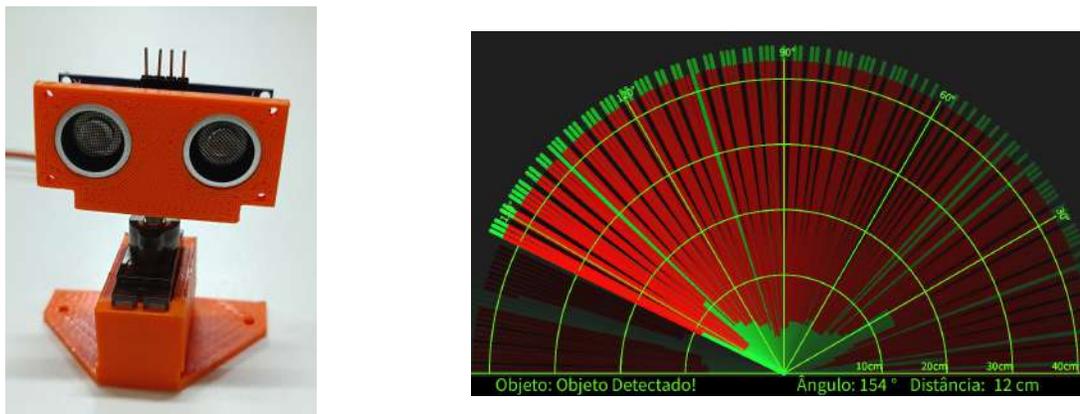


Figura 3: Esquerda: Montagem do sonar. Direita: Visualização do funcionamento do sonar no ambiente Processing.

7 Resultados

Os resultados obtidos no final da construção do sonar foram muito satisfatórios e motivadores para todos os envolvidos na atividade. A Figura 3 mostra um dispositivo robusto e funcional, onde cada componente foi cuidadosamente integrado para garantir o funcionamento eficiente do sistema. A precisão da montagem, incluindo as peças impressas em 3D para o suporte do sensor HC-SR04 e do motor micro servo, contribuiu significativamente para a estabilidade e precisão das medições.

A imagem do ambiente processing mostra o sonar em pleno funcionamento, mostrando a visualização em tempo real das medidas de distância obtidas pelo sensor. Este resultado não apenas valida o funcionamento correto do dispositivo, mas também destaca a capacidade dos alunos do Ensino Básico para aplicar os princípios do som, estudados na disciplina de Física e Química, na construção de um pequeno sonar funcional.

A capacidade de traduzir conceitos teóricos numa aplicação prática tangível consolida as aprendizagens dos alunos e motiva-os a explorar e desenvolver as suas aptidões nas áreas STEM. O projeto de construção de um sonar proporcionou aos alunos uma imersão completa no processo de desenho, construção e testes de um dispositivo eletrônico real.

8 Discussão dos resultados e sugestões de melhoria

Este projeto veio destacar o potencial que as atividades práticas e interdisciplinares têm no enriquecimento do processo educativo, proporcionando aos alunos uma experiência de aprendizagem significativa e relevante. Adicionalmente, este projeto sugere oportunidades para futuras explorações e projetos no campo das STEM.

O conhecimento que os alunos têm da existência de uma relação entre a velocidade de propagação do som e a temperatura do ar, levou-os à sugestão de uma possível melhoria no projeto. Os alunos propuseram a inclusão de uma etapa inicial para medir a velocidade do som no ar, utilizando o próprio sonar. Bastaria para isso colocar um objeto a uma distância fixa do sensor e reescrever o código para ser possível calcular a velocidade do som. Seria útil considerar ainda a inclusão de um termómetro digital no projeto.

9 Conclusão

Este trabalho demonstra a viabilidade de envolver alunos do Ensino Básico em projetos práticos e interdisciplinares, nomeadamente na construção de um sonar de muito baixo custo e funcional. Ao longo desta jornada, os participantes construíram um dispositivo real e expandiram os seus conhecimentos de física, eletrónica, programação, modelagem e impressão 3D. Cada etapa do projeto proporcionou uma oportunidade de aprendizagem e descoberta.

10 Referências

- [1] Zimmerman, D. (2002). Paul Langevin and the Discovery of Active Sonar or Asdic. *The Northern Mariner Le Marin Du Nord*, 12(1), 39-52.
- [2] Majid et al. (2015). Ultrasonication and food technology: A review. *Food Science & Technology*, 2-5.
- [3] Pingret et al. (2012). F. Lab and pilot-scale ultrasound-assisted water extraction of polyphenols from apple pomace. *Journal of Food Engineering*, 111, 73-81.
- [4] Mason et al. (1996). The uses of ultrasound in food technology. *Ultrasonics Sonochemistry*, (3), S253-S260.