

Robot controlado por som

Nos dias de hoje é cada vez mais comum a criação de *robots* que consigam executar tarefas repetidamente e com eficiência. O enquadramento deste projeto prende-se com a inclusão de som em *robots*, no sentido de influenciar a sua locomoção através de sons. Assim, este projeto de cariz académico tem como objetivo desafiar os alunos no sentido de desenvolverem soluções robóticas que interajam com os humanos encontrando pontos de convergência tecnológica e novos paradigmas, com recursos relativamente baixos e contemplando soluções modulares e *open source*.

1. INTRODUÇÃO

O artigo aborda o desenvolvimento de um *robot* com 2 tipos de controlo, um através de som (palmas) e outro, mais comum, através de 2 sensores de infravermelhos. Este projeto foi realizado no curso de Técnico Eletrónica, Automação e Comando do INETE [1]. Um dos grandes objetivos do projeto consiste na tradução do som do bater de palmas para instruções de controlo e conseqüente locomoção do *robot*. O outro grande objetivo é o *robot* ser de tamanho reduzido e de baixo consumo. Assim, o *robot* desenvolvido possui 2 modos de funcionamento: modo autónomo e modo manual. No modo autónomo a sua locomoção está dependente, exclusivamente dos 2 sensores de infravermelhos, permitindo uma deslocação da seguinte forma: frente, trás, esquerda, direita e parado. No modo manual, o *robot* é controlado em função de sons (tons) existentes no seu meio envolvente, no caso que se está a explorar através de palmas.

2. ARQUITETURA DO ROBOT

A arquitetura desenvolvida para o *robot* está representada através da Figura 2. Esta consiste num Arduino Nano, responsável pela aquisição de dados dos 3 sensores incluídos no *robot* e de acordo com as instruções de programação (em linguagem C/C++) implementadas e resistentes na sua memória FLASH. Estão incluídos no projeto 3 sensores, 2 de infravermelhos, para a deteção de obstáculos existentes no meio envolvente e um

microfone de elétrodos, que está a funcionar como sensor de som, com o objetivo de detetar o som das palmas. A locomoção é assegurada por um *drive* de potência L293, em circuito integrado e com arquitetura em ponte H, que fornece potência para os 2 motores DC. A alimentação do *robot* é assegurada por uma pilha de 9 VDC. Por fim, para a programação do Arduino utilizou-se a linguagem C/C++ devido à sua simplicidade e à sua aplicação em componentes eletrónicos.

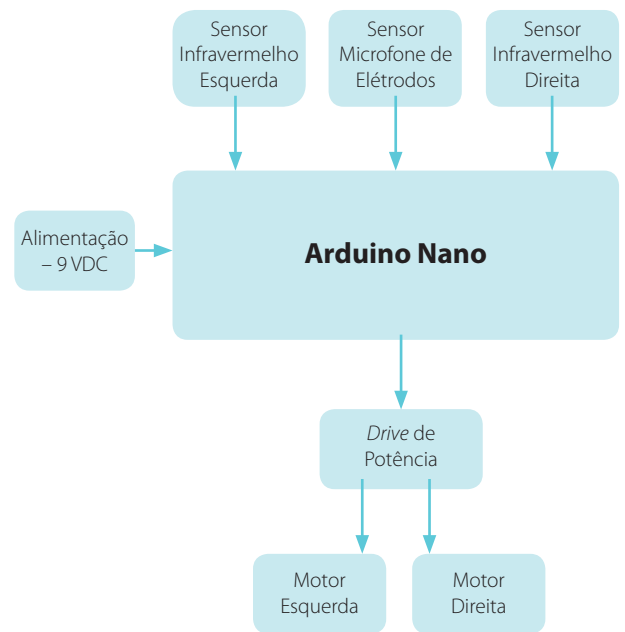


Figura 2. Arquitetura do robot.

2.1. Microcontrolador – Arduino Nano

O Arduino Nano é uma placa pequena, completa e baseada no microcontrolador ATmega 328 (para placas Arduino Nano versão 3.x – utilizado neste projeto). Este Arduino é alimentado com um cabo USB Mini-B em vez de um padrão. O seu tamanho reduzido faz desta placa uma boa opção para este projeto, tornando-o bastante pequeno e compacto. Possui 14 pinos digitais de *input/output* dos quais 6 podem ser usados como pinos de saída para sinais *Pulse Width Modulation* (PWM), 6 pinos de



Figura 1. Aspetto físico do robot.

input analógicos, um cristal oscilador a 16 MHz, uma porta USB que possibilita a comunicação com um computador, uma entrada de alimentação e um interruptor de *reset*. Tal como outras soluções Arduino, esta placa contém todo o *hardware* necessário para o funcionamento do microcontrolador ATmega328.

Os microcontroladores surgiram para servirem de base para sistemas embebidos, ou seja sistemas que trabalhem sem interrupção e sem intervenção humana, sendo por isso uma ferramenta muito útil para controlar algo com baixos recursos [2]. Os microcontroladores possuem uma grande flexibilidade na criação de *software* e no desenvolvimento do *hardware* que o rodeia, usufruindo da comunicação entre ambos. Um microcontrolador tal como um computador é construído com vários componentes (módulos), que enquanto num computador se encontram separados e à vista podendo ser substituídos, num microcontrolador estes módulos encontram-se concentrados num pequeno encapsulamento, assegurando assim o funcionamento básico do microcontrolador. Os módulos que o compõem são a unidade de memória, Unidade Central de Processamento (CPU), BUS, portos de *input/output*, porta de comunicação série, *timers*, *Analog to Digital Converter* (ADC) [3].

A arquitetura do microcontrolador é constituída por 10 blocos. A unidade de processamento central (CPU) é responsável por todo o processamento de dados do microcontrolador. Esta unidade é responsável por interpretar os comandos, faz a leitura de dados e ativa as portas de I/Os ou periféricos (I2C, Temporizador, entre outros) se necessário. As portas I/O de um microcontrolador são responsáveis pela “*entrada*” e “*saída*” do mesmo. No microcontrolador estas portas são um dos componentes essenciais para a comunicação do microcontrolador com qualquer componente externo. O ADC é um componente extremamente importante que converte sinais (V) analógicos para sinais digitais. Sendo um conversor de 10 *bits* tem a capacidade de detectar 1,024 (2^{10}) níveis analógicos discretos, usado no projeto para a aquisição dos sinais oriundos dos 3 sensores. O I2C (Circuito Inter-integrado) é um BUS série multimestre desenvolvido pela Philips que é usado para ligar periféricos de baixa velocidade, por exemplo sensores, no sentido de reduzir a complexidade do circuito. A Unidade Série ou *Universal Synchronous Asynchronous Receiver Transmitter* (USART) é um formato padrão para a comunicação de dados em série de forma assíncrona, dois fios são usados para transmitir dados, um em cada direção, em regime *full-duplex*, ou seja, totalmente bidirecional. Cada dispositivo tem de ter o seu próprio *clock* e as velocidades devem ser iguais. Em forma síncrona, uma das entidades é mestre enquanto a outra é escravo. O temporizador é um dispositivo capaz de medir o tempo, sendo um tipo de relógio especializado. Este pode ser usado para controlar a sequência de um evento ou processo. Na arquitetura de circuitos microcontroladores, os temporizadores são utilizados para gerar bases de tempo que podem ser utilizadas para os mais diversos fins, como por exemplo gerar sinais *clock* para outros periféricos do *chip*, calcular intervalos de tempo ou medir período de sinais, de elevada importância para gerar sinais PWM. Em termos de memórias existem 3 tipos distintos, descritas nos próximos parágrafos.

A EEPROM/FLASH (*Electrically-Erasable Programmable Read-Only Memory*) é um tipo de memória não-volátil (há persistência de dados), usada em computadores e outros dispositivos eletrónicos para armazenar pequenas quantidades de dados

“

A realização deste projeto permitiu verificar a possibilidade de se controlar um *robot* através de som, neste caso o bater de palmas, sem recurso a outro tipo de periférico, por exemplo, comando. Foi possível eliminar por *hardware* e *software* todo o ruído envolvente ao *robot*, com a exceção das frequências que o bater de palmas geram.

que precisam de ser guardados quando a energia é removida. Apesar do nome conter *READ-ONLY* ou ler apenas, também tem a capacidade de podermos escrever ou *WRITE*. Na arquitetura da placa Arduino Nano, cerca de 500 *byte* desta memória são para o *bootloader*. A memória ROM (*Read-Only-Memory*) é um tipo de memória que permite apenas a leitura, ou seja, as suas informações são gravadas pelo fabricante uma única vez e após isso não podem ser alteradas ou apagadas. São memórias cujo conteúdo é gravado permanentemente. A memória RAM (*Random-Access-Memory*) é um tipo de memória que permite a leitura e a escrita, utilizada como memória primária em sistemas em sistemas eletrónicos digitais.

O BUS é um barramento que possibilita a comunicação entre componentes ou entre entidades, levando a informação de um entidade para todas as outras.

2.2. Sensores infravermelhos e microfone de elétrodos

Os sensores são dispositivos que a partir de uma ação exterior conseguem apresentar aos seus terminais uma diferença de potencial ou variação de impedância, que pode ser de natureza indutiva, capacitiva ou resistiva. Os sensores incluídos no *robot* são dois sensores infravermelhos (fabricante Sharp GP2Y0A41SK0F), sendo sensores ativos e de curto alcance (4 cm a 30 cm), e têm uma saída analógica de tensão. O objetivo destes sensores passa por detetar os obstáculos no ambiente envolvente ao *robot*. Para a deteção de sons utilizou-se um microfone de elétrodos e um circuito de acondicionamento de sinal, com o objetivo de tornar legível o sinal por parte do microcontrolador ao nível da frequência e tensão. O circuito para acondicionar o sinal oriundo do microfone, inclui um amplificador operacional (AmOp), garantindo uma amplificação de tensão de 100 vezes, do sinal sonoro que o microfone tenha, por exemplo, bater de portas, bater de palmas, entre outros. Este sinal analógico é depois convertido para digital a 10 *bits*, utilizando o ADC do microcontrolador residente no Arduino Nano. O mesmo acontece com os 2 sensores infravermelhos.

2.3. Pré-atuadores, atuadores e sistema de alimentação

Para garantir a locomoção do *robot*, para além dos dois motores DC, torna-se fundamental a inclusão de um *drive* de potência. O *drive* de potência utilizado é o circuito integrado L293 que assegura a potência suficiente para os motores DC utilizados, mas também permite regular, através de sinais PWM, a velocidade e direção de cada motor. O L293 tem 6 pinos específicos para controlar a direção e velocidade dos 2 motores, 3 para cada um deles. Dos 3 pinos, 2 são usados para o controlo da direção, onde recebem a informação binária de 0 ou 1 para alimentar ou não cada terminal do motor. O outro pino controla a velocidade,

através de um sinal PWM gerado e controlado pelo Arduino. O PWM varia entre 0 e 256 (*timer 0* do microcontrolador que tem 8 *bits*) e através desta faixa de valores consegue-se controlar a velocidade, em que 0 é parado e 256 é a velocidade máxima. O motor DC utilizado na realização deste projeto foi o servo Hitec HS-422. Este servo é capaz de fornecer a 6 V uma força de aproximadamente 57 oz-in de binário máximo a 0,16 segundos/60°. Os servos são dispositivos que recebem um sinal de controlo, verificam a posição atual, atuam no sistema indo para a posição desejada, e estão limitados na sua rotação a um determinado ângulo. Neste projeto são utilizados dois servos modificados que consistem num motor DC onde se retirou a *drive* de potência interno, e ainda uma modificação mecânica, permitindo que rode a 360 graus.

O sistema de alimentação do *robot* é através de uma pilha de 9V.

3. ALGORITMOS DESENVOLVIDOS

A Figura 3 representa o algoritmo geral implementado no *robot*, recorrendo à linguagem de programação C/C++.

Após a realização do projeto verificou-se, na prática, a tradução do bater das palmas para os diversos movimentos que o *robot* executa, ou seja, o utilizador bate palmas e consoante o número de palmas o *robot* movimenta-se. Assim, com o bater de uma palma o carro move-se para a frente, com 2 palmas

move-se para trás, com 3 palmas vira à esquerda, com 4 palmas vira à direita, com 5 palmas pára e com 6 palmas move-se de forma autónoma, utilizando os seus sensores infravermelhos para se desviar dos obstáculos. Devido às interferências entre os 2 tipos de sensores utilizados (infravermelhos e de som) foi necessário filtrar a alimentação, e para isso foi usado um filtro passivo RC (passa baixo) e não um filtro ativo pois o filtro RC (resistência condensador) é mais elementar e abrangeu a mesma frequência de corte mais elevada do que a calculada, inibindo parcialmente o ruído.

4. CONCLUSÃO

A realização deste projeto permitiu verificar a possibilidade de se controlar um *robot* através de som, neste caso o bater de palmas, sem recurso a outro tipo de periférico, por exemplo, comando. Foi possível eliminar por *hardware* e *software* todo o ruído envolvente ao *robot*, com a exceção das frequências que o bater de palmas geram.

5. REFERÊNCIAS

[1] www.inete.pt.
 [2] Pires, L., *Sistemas Embebidos*, Sistemas Digitais, INETE, março 2014, Portugal.
 [3] Pires, L., *Microcontroladores AVR*, Sistemas Digitais, INETE, setembro 2014, Portugal. ↗

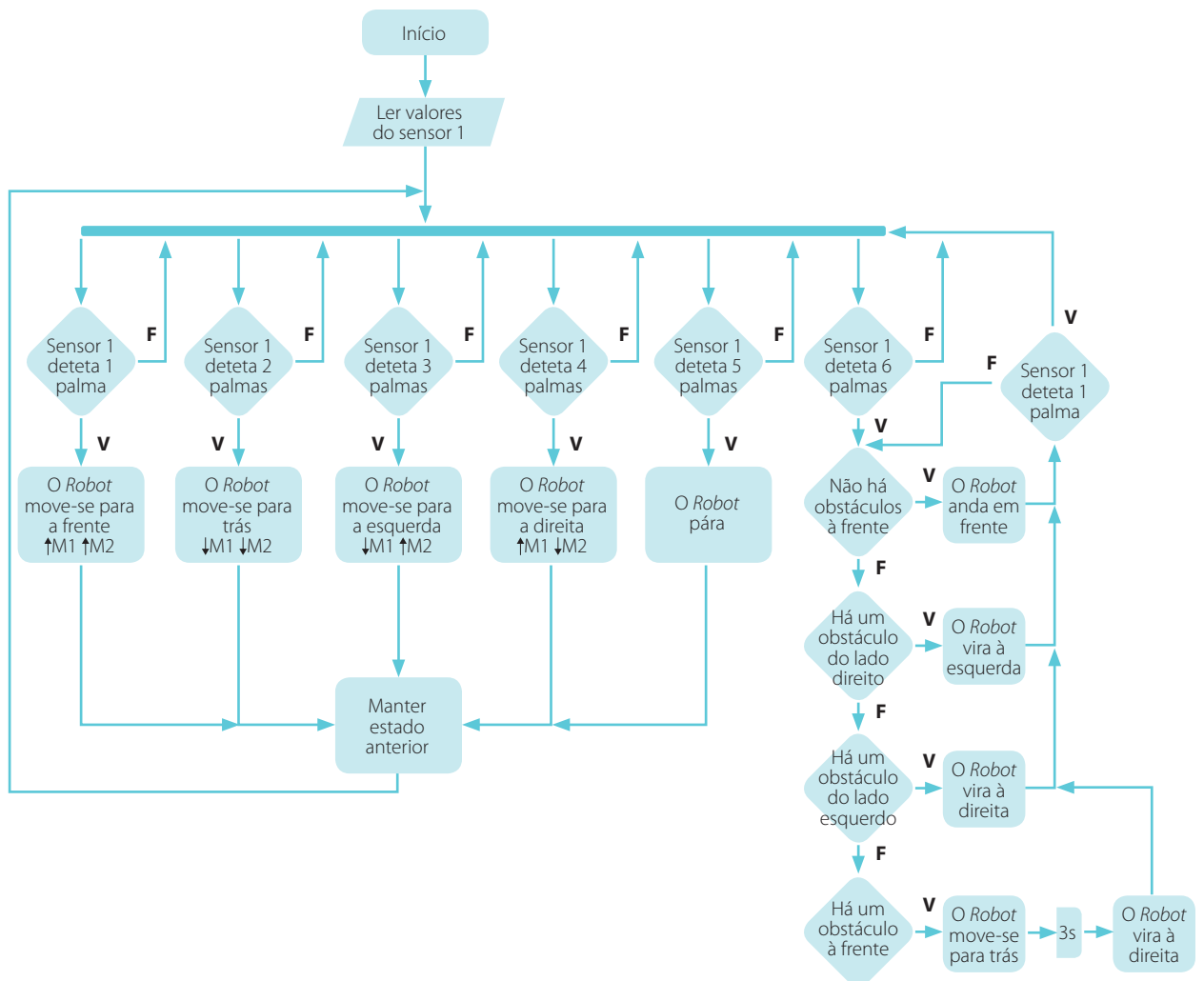


Figura 3. Algoritmo geral do *robot*.