

Crab-Robot

Hoje em dia, com o avanço da tecnologia, a robótica cada vez mais começa a abandonar o uso de rodas e progride para outros tipos de locomoção, como a locomoção por pernas ou patas.

Este tipo de locomoção contém vantagens em relação a rodas, como por exemplo proporcionar uma maior manobrabilidade em terrenos irregulares e em desníveis acentuados, e a possibilidade de ultrapassar um obstáculo sem ter de o contornar. Por outro lado possui várias desvantagens como a impossibilidade de igualar a velocidade e simplicidade de um sistema a rodas, o menor número de componentes e não é necessário existir um grande controlo ao nível do equilíbrio do *robot*.

1. INTRODUÇÃO

O artigo aborda o desenvolvimento de um *robot* com locomoção através de patas, semelhante a um caranguejo. Este foi realizado no curso de Técnico Eletrónica, Automação e Comando do INETE [1]. Este *robot* movimenta-se de forma autónoma em função do ambiente circundante, podendo também obedecer a comandos remotos através de uma ligação sem fios Bluetooth. A aplicação para controlo remoto funciona num *smartphone* de sistema operativo Android. O *robot* dispõe de seis patas articuladas que lhe permitem movimentos verticais e horizontais em três eixos, permitindo a deslocação em qualquer direção, bem como ultrapassar pequenos obstáculos, ou afastar-se de buracos existentes no seu percurso. Possui uma garra articulada para captura de presas. Para a deteção de obstáculos utiliza-se um conjunto de sensores ultrassons, dispostos de forma a cobrirem 360° na horizontal. Para que possa reagir a aproximações de objetos em cima do *robot*, dispõe de um sensor de ultrassons adicional, colocado perpendicularmente em relação aos restantes. Possui detetores de luminosidade, que permitem detetar locais escuros onde se possa esconder, tal como o comportamento de um caranguejo.

Para possibilitar ao *robot* medir a inclinação do plano onde se desloca, foi colocado um acelerómetro, que deteta variações nos três eixos, x, y e z, permitindo que o *robot* se desloque de forma estável e evite inclinações excessivas.

Assim, o projeto é constituído por 10 sensores de ultrassons que detetam obstáculos numa amplitude de 360°.

corpo com base de alumínio, as partes articuladas e de suporte de sensores são feitas de PLA, tendo sido modeladas e impressas em 3D.

2. ARQUITETURA DO ROBOT

A arquitetura desenvolvida para o *robot* está representada através da Figura 2.

2.1. Microcontroladores

Os microcontroladores [2] são unidades de processamento de dados e contém: a CPU (unidade central de processamento de dados), memórias permanentes de leitura (ROM) do fabricante, memórias voláteis de leitura/escrita (RAM). As memórias temporárias têm de estar constantemente alimentadas de modo a não perderem os dados armazenados. No caso das memórias permanentes alteráveis, tais como por exemplo FLASH, EEPROM, EPROM, sendo nestas alojado o programa da aplicação criada, contém também pinos de entrada e saída e pinos de alimentação. Os microcontroladores têm uma de duas arquiteturas: a arquitetura Von-Neumann ou a arquitetura Harvard. Na arquitetura Von-Neumann todos os elementos do

Com um sensor de ultrassons adicional é possível detetar objetos em cima do *robot* e, utilizando quatro dos 10 sensores de ultrassons, identificam-se desníveis. Os quatro sensores de luminosidade LDR detetam a presença do *robot* em locais escuros. Quando deteta uma zona escura o *robot* desloca-se para o local e mantém-se lá por algum tempo. Utiliza 22 micro-servos para a sua locomoção. Tem um módulo de comunicação Bluetooth para controlo remoto através de uma *app*. Utiliza duas baterias LiPo de 3,7 V 7400 mAh e um sistema de monitorização de baterias com indicadores de estado de bateria a LED. O *robot* tem um

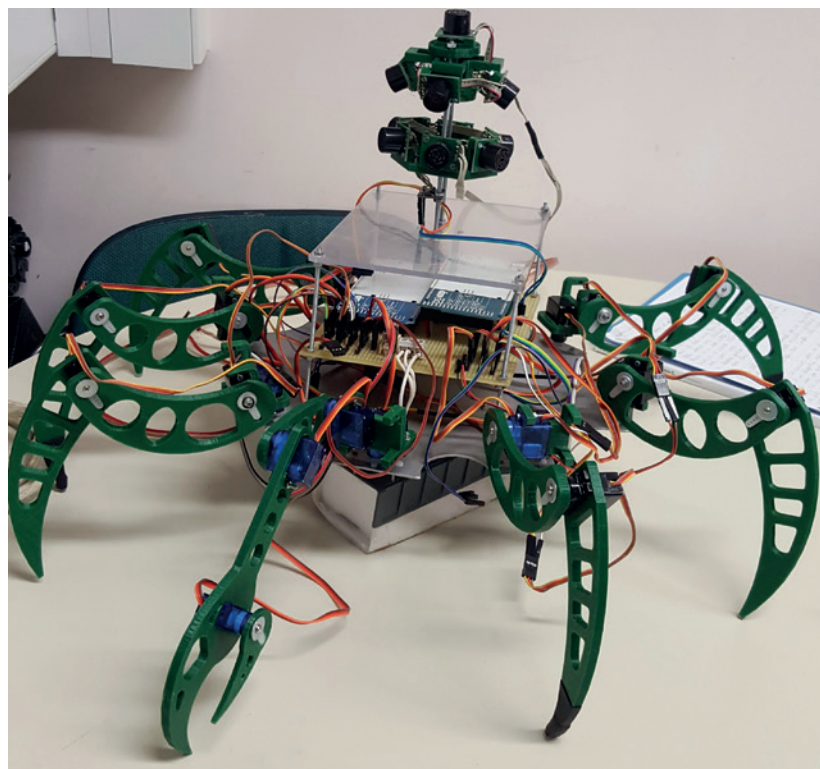


Figura 1. Aspeto físico do *robot*.

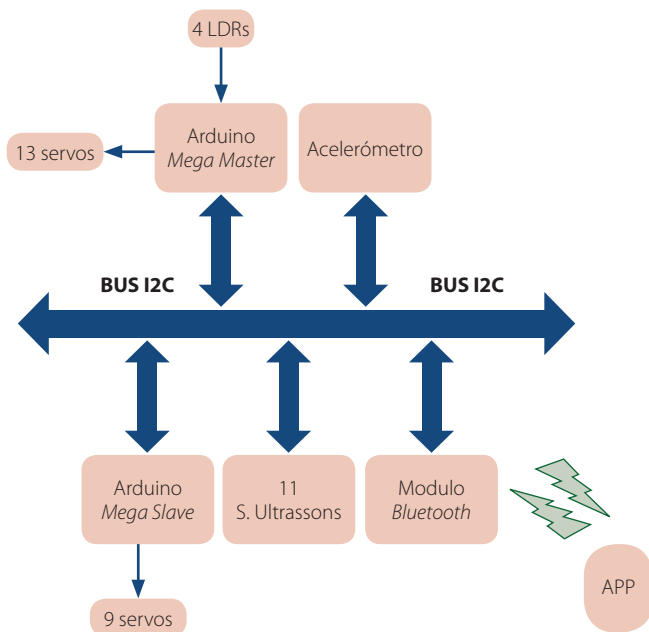


Figura 2. Arquitetura do robot.

microcontrolador estão alinhados para que os dados sejam processados sem apresentar erros. Com esta arquitetura de microcontrolador a máquina digital armazena os programas na mesma unidade de memória que os dados, pode-se assim manipular os programas e utilizar apenas uma unidade de memória e uma de processamento para comportar os dados e as instruções. A arquitetura Harvard é derivada da arquitetura Von-Neumann, foi criada pela necessidade de aumentar a rapidez de processamento de dados por parte do microcontrolador. Desta forma, na arquitetura Harvard a unidade de processamento de informação está no centro. São utilizadas duas memórias, memória de instruções e memória de dados, sendo possível aceder às duas simultaneamente. No projeto utiliza-se o microcontrolador ATmega2560, na *board* Arduino Mega 2560 [3]. Os Arduínos Mega 2560 estão ligados entre si através de uma rede I2C, isto para se poder utilizar as 22 saídas PWM e consegue-se controlar todos os microservos do *robot*, não é possível controlar apenas com um Arduino Mega pela limitação de 15 saídas analógicas PWM.

2.2. Sensores

Para que os dispositivos eletrônicos, *robots* e automatismos realizem as suas funções é necessário que estes tenham a capacidade de interagir com o ambiente que os rodeia, para isto utilizam-se sensores. Os sensores são dispositivos que convertem grandezas físicas que se pretendem medir, em grandezas elétricas capazes de

serem interpretadas por sistemas eletrônicos. Internamente dividem-se em duas partes: primeiro, o elemento que sofre alterações físicas na presença da unidade que se pretende medir; o segundo, o transdutor, que transforma essas alterações físicas em impedâncias (resistivas, capacitivas, indutivas). É possível agrupar os sensores em dois grupos, de acordo com as características do sinal retornado, sensores passivos e sensores ativos. Nos sensores passivos a variação na medida a realizar demonstra-se numa alteração da sua impedância, são exemplos sensores resistivos, capacitivos e indutivos. Os sensores deste grupo utilizados no projeto são os quatro sensores LDR. Os sensores ativos comportam-se como um gerador, aproveitando a energia proveniente do processo a medir, como por exemplo sensores piezoelétricos. Este sensor quando sofre alterações físicas, como por exemplo vibrações, produz uma diferença de potencial nos seus terminais gerando desta forma energia, como por exemplo os sensores de ultrassons.

Utilizam-se 11 sensores ultrassônicos SRF02, um acelerômetro ADXL345 e quatro sensores de luminosidade LDR. Os sensores ultrassônicos SRF02 medem distâncias entre 16 cm a 6 m, através da emissão de sinais de ultrassons a 40 kHz. Esta distância é calculada através do tempo que demora um impulso a chegar ao sensor após ter sido transmitido o impulso de ultrassom. Estes sensores são alimentados a 5 V e ligam-se diretamente à rede. Este sensor pode ser calibrado pelo

utilizador, apesar de não ser necessário pois o sensor é calibrado automaticamente sempre que é alimentado.

Na rede I2C estes sensores podem ter 16 possíveis endereços, E0, E2, E4, E6, E8, EA, EC, EE, F0, F2, F4, F6, F8, FA, FC ou FE.

O sensor ADXL345 é um acelerómetro de três eixos, x, y, z, que mede em força G. Este módulo é alimentado a 5 V e liga-se à rede I2C para comunicar com o microcontrolador. O endereço padrão do módulo é 0x53.

Os sensores de luminosidade LDR são resistências que variam a sua impedância resistiva com a intensidade luminosa incidente na placa do sensor. Estes sensores estão ligados às entradas analógicas do Arduino Mega 2560 através de um circuito de acondicionamento de sinal (divisor de tensão) para adaptar a variação de impedância para um sinal de tensão que o microcontrolador consegue interpretar.

2.3. Atuadores

Os atuadores são dispositivos que têm a capacidade de converter energia elétrica em energia não elétrica, ou seja, numa ação mecânica, sinalizadora ou força. Um exemplo de atuador no projeto é um microservo. Utilizam-se como atuadores 22 microservos. Os microservos são alimentados entre 4,8V e 6V, consomem menos que 500 mA, e são controlados por sinais PWM que os microcontroladores utilizados produzem. Estes sinais PWM vão controlar a posição do veio do servo, que pode ser entre 0° a 180° de rotação.

2.4. Sistemas de comunicação

O protocolo I2C é um protocolo de comunicação a curta distância e banda estreita, tem elevada facilidade de integração de vários dispositivos entre si, entre 112 dispositivos, num barramento do tipo BUS série, e é retro compatível com todas as suas evoluções tecnológicas. O funcionamento do BUS I2C é do tipo *Master-Slave*, mas suporta múltiplos *Masters* e múltiplos *Slaves*, e tanto o *Master* como o *Slave* podem transmitir e receber dados em *half-duplex*. Este protocolo tem três modos de funcionamento:

- *Standard Mode* (0-100kbps), 7 bit endereçamento;
- *Enhanced Mode* (0-400kbps), 7-10 bit endereçamento;
- *High Speed Mode* (0-3.4 Mbps), 7-10 bit endereçamento. O I2C é composto por duas linhas que formam o BUS:

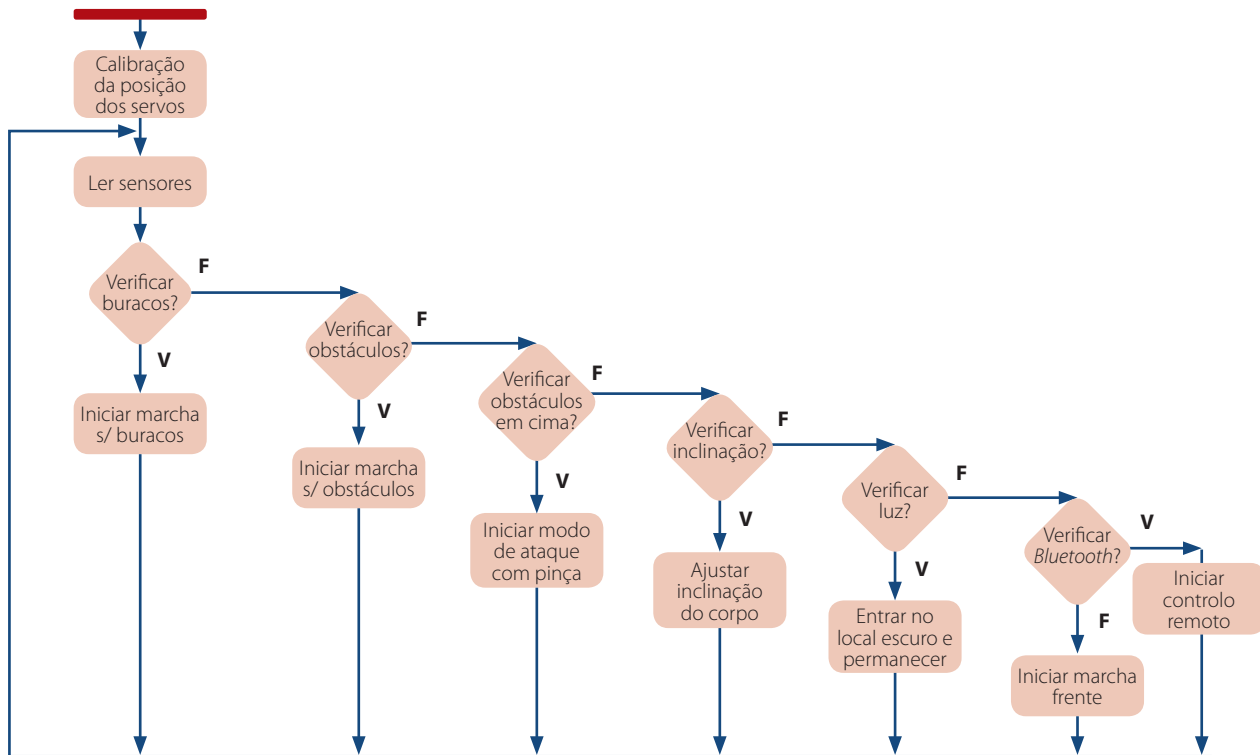


Figura 3. Algoritmo geral do robot.

- SDA: *Serial Data Line* (dados bidirecionais);
- SCL: *Serial Clock Line* (clock);

Ambas as linhas são ligadas a 5 V através de resistências de 2,2 k Ω que fazem um *Pull-Up* ao sinal, visto que os dispositivos têm saídas de coletor aberto.

Para existir uma transmissão de dados numa rede I2C entre um dispositivo e outro, é preciso que o dispositivo emissor envie uma condição de *start*. Nesta etapa, o SCL mantém-se num estado *High* enquanto o SDL desce para um estado *Low*. De seguida é enviado o endereço do dispositivo que recebe os dados compostos por sete *bits*, começando pelo MSB. O recetor gera um ACK, indicando que foram recebidos os sete *bits* anteriores. O emissor começa a transmitir os dados (*Master* ou *Slave*), *byte* a *byte*. No final de cada *byte*, o emissor (*Master* ou *Slave*) gera um novo ACK. Estas etapas de nove *bits* (oito dados e um ACK) são repetidas até não haver dados para transmitir. No final, numa transação de escrita, quando o recetor dos dados receber todos os que foram requisitados, este fica à espera de um último ACK e gera a condição de *stop*. No caso de uma transação de leitura o *Master* não envia ACK pelo último *byte* que recebe, isto indica ao *Slave* que pode parar de transmitir dados. Quando o *Slave* parar, o *Master* envia a condição de *stop*.

2.5 Algoritmos desenvolvidos

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

2.6. Locomoção de um robot com seis patas

Um *robot* com seis membros para locomoção é um dos tipos de *robots* que consegue estar mais adaptado a condições mais desfavoráveis para o movimento. Estes *robots* podem obedecer aos seguintes tipos de marcha:

- Marcha tripé;
- Marcha quadrúpede;
- Marcha cinco por um;
- Marcha livre.

Na marcha tripé, o *robot* movimenta-se em dois conjuntos de três membros: um conjunto é constituído pelas patas 1, 3 e 5, estas mexem-se de forma síncrona; o outro conjunto é constituído pelas patas 2, 4 e 6, mexendo-se igualmente de forma síncrona. O primeiro conjunto inicia a marcha para a frente e o outro conjunto suporta o corpo no chão, e depois deste movimento acabar o segundo conjunto inicia a marcha para a frente enquanto o primeiro conjunto suporta o corpo no chão. Este tipo de marcha tende a ser mais rápido que outras marchas. Na Figura 4, representa-se o diagrama de posicionamento de patas num *robot* retangular.



Figura 4. Diagrama de posicionamento de patas num *robot* retangular.

3. CONCLUSÃO

Estes tipos de locomoção contêm vantagens em relação às rodas tais como, proporcionarem uma maior manobrabilidade em terrenos irregulares e em desníveis acentuados, assim como a possibilidade de ultrapassar um obstáculo sem ter de o contornar. Estes também contêm desvantagens como, por exemplo, não conseguirem igualar a velocidade e simplicidade de um sistema a rodas, e isto foi notório na construção e no *design* da estrutura física. O projeto apresentado neste artigo teve uma primeira versão como projeto de uma Prova de Aptidão Profissional em 2015 [4].

4. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] <http://www.inete.pt>
- [2] Pires, L., *Microcontroladores AVR*, Sistemas Digitais, INETE, setembro 2014, Portugal.
- [3] <http://arduino.cc/en/Main/ArduinoBoardMega>
- [4] Marques, D., *Crab-Robot*, Prova de Aptidão Profissional, INETE, 2015. 📄