

Desenvolvimento de recursos educacionais integrados numa abordagem STEM utilizando kits de robótica móvel

Rodrigo Mendes Antunes

Dissertação para obtenção do Grau de Mestre em
Engenharia Eletromecânica
(2º ciclo de estudos)

Orientador: Prof. Doutor Pedro Miguel de Figueiredo Dinis Oliveira Gaspar
Co-orientador: Mestre Martim Lima de Aguiar

junho de 2023

Declaração de Integridade

Eu, Rodrigo Mendes Antunes, que abaixo assino, estudante com o número de inscrição M11620 de/o Engenharia Eletromecânica da Faculdade de Engenharia, declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o **Código de Integridades da Universidade da Beira Interior**.

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, que em particular atendi à exigida referenciação de frases, extratos, imagens e outras formas de trabalho intelectual, e assumindo assim na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 12/06 /2023

Rodrigo Mendes Antunes

(assinatura conforme Cartão de Cidadão ou preferencialmente
assinatura digital no documento original se naquele mesmo formato)

Agradecimentos

Desejo expressar o meu maior agradecimento ao meu orientador, Professor Doutor Pedro Miguel de Figueiredo Dinis Oliveira Gaspar. Pela confiança e sabedoria que sempre transmitiu ao longo do trabalho desenvolvido nesta dissertação, por todas as oportunidades que me permitiram crescer enquanto pessoa e profissional.

Um agradecimento muito especial ao meu Co-orientador, Professor Martim Lima de Aguiar. Pela amizade, por todas as ideias, discussões construtivas e pela disponibilidade exibida desde o primeiro dia.

Agradecer aos meus pais e irmão, que sempre me apoiaram e acreditaram em mim não só ao longo da dissertação, mas em todo o percurso académico. Sem o trabalho, esforço e dedicação deles nunca seria possível.

Aos meus colegas e amigos, em especial ao Daniel Ferreira, Mariana Veloso, Daniel Nunes, Pedro Pereira, Carlos Marques, Miguel Cruz, Bruno Eusébio e ao Gonçalo Almeida pelo inestimável apoio e ajuda prestada em todas as fases, pela confiança e companheirismo.

Ao projeto UBICAR, por me ter permitido crescer, viajar e fazer amizades para a vida.

Ao Sr. Nuno Santos e Mestre João Correia, técnicos do FabLab da Universidade da Beira Interior pelo apoio prestado nesta fase final.

A todos os que direta ou indiretamente contribuíram para a realização desta dissertação.

Resumo

A Indústria 4.0 trouxe consigo uma onda de tecnologias de ponta, como a Fabrico Aditivo, Inteligência Artificial, Realidade Aumentada, Sistemas Ciberfísicos, Blockchain, Cibersegurança e muitas outras. Estas inovações têm transformado diversos setores, capacitando as empresas a enfrentarem desafios como as flutuações imprevisíveis na procura e a volatilidade do mercado. No entanto, a Indústria 4.0 enfrenta um obstáculo significativo em termos de questões relacionadas com recursos humanos. Para operar efetivamente neste ambiente evoluído, os funcionários precisam de formação adequada, o que exige a implementação de práticas de gestão eficazes. Isto assegura que os trabalhadores qualificados possam adaptar-se às exigências das fábricas inteligentes. Ao estabelecer políticas favoráveis, as organizações podem aprimorar as suas capacidades dinâmicas, fomentando uma cultura inovadora que impulsiona o progresso contínuo na indústria. Utilizar kits de robótica móvel numa abordagem STEM (*Science, Technology, Engineering and Mathematics* - Ciência, Tecnologia, Engenharia e Matemática) pode proporcionar uma experiência de aprendizagem mais envolvente e prática para os estudantes, ajudando-os a desenvolver competências em programação, resolução de problemas e pensamento crítico. Além disso, a integração de recursos educacionais nesta abordagem pode ser uma forma eficaz de envolver os estudantes em atividades práticas e estimular o seu interesse pelas disciplinas STEM. Este trabalho envolve o desenvolvimento de protocolos laboratoriais para diferentes níveis académicos, numa abordagem sequencial que aprofunda o conhecimento STEM, com o objetivo de ensinar robótica utilizando kits de robótica móvel. O desenvolvimento de código e complementos para os protocolos laboratoriais e a realização de testes num ambiente de sala de aula também estão incluídos. O processo de ensino/aprendizagem é avaliado através de inquéritos regulares aos estudantes. No geral, este trabalho tem como objetivo fornecer uma abordagem pedagógica prática e interdisciplinar para o ensino de robótica utilizando kits de robótica móvel, com foco no desenvolvimento de competências STEM em estudantes de diferentes níveis académicos. Esta abordagem visa promover a aquisição de conhecimentos em tópicos fundamentais para o futuro da engenharia.

Palavras-chave

Recursos educacionais integrados, Robótica, Kits de robótica móvel, Educação STEM.

Abstract

Industry 4.0 has ushered in a wave of cutting-edge technologies like Additive Manufacturing, Artificial Intelligence, Augmented Reality, Cyber-physical systems, Blockchain, Cybersecurity, and many others. These breakthroughs have transformed diverse sectors by empowering businesses to address challenges such as unpredictable shifts in demand and market volatility. However, Industry 4.0 faces a notable hurdle in terms of human resource issues. To effectively operate in this advanced landscape, employees need adequate training, necessitating the implementation of effective management practices. This ensures that these knowledgeable individuals can adapt to the requirements of intelligent factories. By establishing favourable policies, organizations can enhance their dynamic capabilities, fostering an innovative culture that propels further progress in the industry. Using mobile robotics kits in a STEM (Science, Technology, Engineering, and Mathematics) approach can provide a more engaging and practical learning experience for students, helping them develop skills in programming, problem-solving, and critical thinking. In addition, the integration of educational resources in this approach can be an effective way to embrace students in practical activities and stimulate their interest in STEM subjects. This work involves the development of laboratory protocols for different academic levels, with a sequential approach that deepens STEM knowledge, aimed at teaching robotics using mobile robotics kits. The development of code and add-ons for laboratory protocols and testing in a classroom setting is also included. The teaching/learning process is evaluated through regular student surveys. Overall, this work aims to provide a practical and interdisciplinary pedagogical approach to teaching robotics using mobile robotics kits, with a focus on developing STEM skills in students at different academic levels. This approach aims to promote knowledge acquisition on topics that are fundamental for the future of engineering.

Keywords

Integrated educational resources, Robotics, Mobile robotics kits, STEM education.

Índice

Agradecimentos	iii
Resumo	v
Abstract	vii
Índice	ix
Lista de Figuras	xi
Lista de Tabelas	xv
Nomenclatura	xvi
1. Introdução	1
1.1. Enquadramento	1
1.2. O problema em estudo e a sua relevância	3
1.3. Objetivos e contribuição da dissertação	4
1.4. Visão geral e organização da dissertação.....	4
2. Estado da Arte	7
2.1. Pensamento Computacional	7
2.1.1 Pensamento Computacional no K-12	8
2.2. Estratégias e metodologias no ensino da programação/robótica	9
2.3. Robótica educacional.....	10
2.3.1 Bee-Bot	10
2.3.2 WeDo 2.0	11
2.3.1. Lego Mindstorms NXT.....	14
2.3.2. mBot	16
2.3.3. Ultimate 1.0	17
2.4. Trabalhos realizados no âmbito da unidade curricular de Robótica Industrial- Primeiro contacto	20
2.4.1. Construção de uma torre de blocos utilizando o Ultimate 2.0	21
2.4.2. Tiro ao alvo com o Ultimate 2.0	24
2.4.3. Resolução de labirinto com mBot2	28
2.5. Nota conclusiva.....	29
3. Materiais e Métodos	31

3.1. Robôs educacionais	31
3.1.1. mBot2	31
3.1.2. Ultimate 2.0	35
3.1.3. Software de programação mBlock 5	40
3.2. Processos de fabrico.....	43
3.2.1. SOLIDWORKS.....	44
3.2.2. Maquinação por Comando Numérico Computorizado (CNC).....	44
3.2.3. Fabrico Aditivo (Impressão 3D)	45
4. Infraestrutura de Apoio à Robótica Móvel	47
4.1. Mesa de trabalho multiusos	47
4.2. Futebot	53
4.3. Garra para mBot2	54
5. Atividades e Recursos Educativos de Robótica	57
5.1. Formação com mBot2	57
5.1.1. Aula 1- Introdução ao mBot2.....	58
5.1.2. Aula 2- Conjugação de sensores	58
5.1.3. Aula 3- Labirinto e criação de variáveis	59
5.2. Universidade de Inverno STEAM	61
5.3. Protocolos desenvolvidos	63
6. Resultados.....	67
6.1. Introdução.....	67
6.2. Questionário 1	68
6.3. Questionário 2.....	68
6.4. Observações durante a realização do Futebot.....	69
6.5. Nota Conclusiva	69
7. Análise de Discussão de Resultados	71
7.1. Formação Escola Profissional	71
7.2. Nota Conclusiva	72
8. Conclusões	73
8.1. Conclusões gerais.....	73
8.2. Sugestões de trabalhos futuros	74
Referências Bibliográficas	75
Anexos	79

Lista de Figuras

Fig. 1. Evolução da instalação de robôs industriais (em milhares de unidades) entre o ano de 2019 e 2021 (IFR, 2022).	2
Fig. 2. Principais áreas da indústria onde ocorreu maior instalação de robôs industriais (em milhares de unidades) entre o ano de 2019 e 2021 (IRF, 2022).	2
Fig. 3. Logo Turtle, o primeiro robô educacional desenvolvido por Seymour Papert (Solomon <i>et al.</i> ,2020)	8
Fig. 4. Botões físicos do Bee-bot (esquerda) e o mapa utilizado na atividade experimental (direita) (Diago <i>et al.</i> , 2022).	11
Fig. 5. Configuração Glowing Snail e respetiva programação.	12
Fig. 6. Configuração Cooling Fan e respetiva programação.	12
Fig. 7. Configuração MoovingSatellite e respetiva programação.	13
Fig. 8. Configuração Frog Robot e respetiva programação.....	13
Fig. 9. Lego Minstorms NXT e o respetivo ambiente de programação Lego NXT (Atmatzidou & Demetriadis, 2016).....	14
Fig. 10. mBot e o respetivo ambiente de programação Scratch.....	16
Fig. 11. Algumas configurações possíveis de montagem com as peças contidas no kit Ultimate 1.0.	18
Fig. 12. Esquema e dimensões do circuito teste proposto pelos investigadores.....	19
Fig. 13. Configuração “robotic arm tank”, uma das configurações base do kit Ultimate 2.0	21
Fig. 14. Montagem do sensor ultrasónico e do sensor infravermelho.....	22
Fig. 15. Comparação entre a posição original da garra e a posição utilizada.....	23
Fig. 16. Localização do sensor fim de curso no braço robótico que previne a colisão com a placa de controlo.....	24
Fig. 17. Localização inicial do Robô e dos Blocos, com a localização onde serão empilhados os blocos	24
Fig. 18. Configuração construída com componentes do kit Ultimate 2.0 e componentes desenvolvidos através da fabrico aditivo.	25
Fig. 19. Identificação dos componentes no sistema pinhão cremalheira.....	26
Fig. 20. Arco desenhado em Solidworks implementado na configuração final do robô.	26
Fig. 21. Exemplos de versões de flechas desenhadas em Solidworks.	27
Fig. 22. Alvo utilizado no projeto tiro ao alvo (esquerda), conjunto arco e flecha desenvolvido em Solidworks (direita).	27

Fig. 23. Labirinto projetado com as respetivas dimensões exteriores (esquerda), coluna que suporta as paredes do labirinto, desenhada em Solidworks, produzida recorrendo fabrico aditivo (direita).....	28
Fig. 24. mBot2 com dois sensores de ultrassons implementados (esquerda), desenho CAD do suporte para os sensores ultrassónicos (direita).	29
Fig. 25. Vista explosiva do mBot2, diferentes componentes presentes na sua constituição (Makeblock, EN mBot2 Getting Started Activities V1.1, 2022)	32
Fig. 26. Sensores para reconhecimento do ambiente, à esquerda o sensor ultrassónico, e à direita o sensor QuadRGB (adaptado de <i>Makeblock, 2022</i>)	33
Fig. 27. Vista explodida do sensor ultrassónico (esquerda), realçando os 8 LEDs azuis na sua constituição e Sensor Quad RGB integrado no mBot2 (direita), realçando os sensores de luz e os LEDs RGB (Adaptado de Makeblock, EN mBot2 Getting Started Activities V1.1, 2022)	34
Fig. 28. Esquema da pista de teste incluída no kit mBot2 (Makeblock, 2022).	35
Fig. 29- Algumas configurações possíveis de construir com as peças disponíveis no kit Ultimate 2.0, desenvolvido pela Makeblock. Na direita, uma impressora 3D desenvolvida a partir das peças no kit, mais um conjunto de add-ons (sensores, display, etc) possíveis de implementar no MegaPi (Makeblock, Ultimate_senior, 2022).	36
Fig. 30- Esquema de especificações técnicas da placa de controlo MegaPi, contida no kit Ultimate 2.0, desenvolvido pela Makeblock. Na direita, a placa adaptadora MegaPi RJ25, utilizada para converter os pinos do MegaPi para a interface RJ25 (Makeblock, Ultimate_senior, 2022).....	36
Fig. 31- Conexão do sensor de linha incluído no kit Ultimate 2.0 com um Arduíno Uno (à esquerda), conexão do sensor de linha com a placa de controlo Orion da Makeblock (à direita) (Makeblock, Ultimate_senior, 2022).	37
Fig. 32- Alguns componentes incluídos no kit Ultimate 2.0, giroscópio, módulo shutter e MegaPi Encode/DC Mux Driver, respetivamente, da Makeblock.	38
Fig. 33- Encoder motor 25 DC 6V/185 RPM, controlado pelo MegaPi Encode/DC Mux Driver (Makeblock, Ultimate_senior, 2022).....	38
Fig. 34- Módulos mBuild e estruturas de suporte desenvolvidos, com o objetivo de expandir as possibilidades de programação dos robôs Makeblock (Makeblock, Ultimate_senior, 2022).	39
Fig. 35. Smart Camera, módulo disponibilizado pela plataforma Makeblock que possibilita a criação de programas que envolvam Visão Computacional e Inteligência Artificial (Makeblock, Smart Camera, 2022).	39
Fig. 36. Ambiente de programação Scratch, disponibilizado pelo software mBlock 5, desenvolvido pela plataforma Makeblock.	40

Fig. 37. Editor Scratch, disponível no software mBlock 5. Subdividido em 3 principais áreas de trabalho: Stage area; Blocks area e Scripts area (Makeblock, mBlock Block-Based Editor-About mBlock5- What is mBlock 5 Block-Based Editor, 2022).	41
Fig. 38. Editor inicialmente desenvolvido pela Makeblock, capaz de transcodificar um programa em Scratch para Python (Makeblock, mBlock Python Editor- About mBlock 5 Python Editor-What is mBlock-Python Editor, 2022).	42
Fig. 39. Transcodificação de um código em Scratch para Arduino, no software mBlock 5. O código manipula os motores instalados no kit Ultimate 2.0, projetado pela plataforma Makeblock.	43
Fig. 40. Versão 1 e 2, respetivamente, do suporte das proteções laterais da mesa dedicada à robótica móvel.	48
Fig. 41. Versão 3 do suporte das proteções laterais da mesa dedicada à robótica móvel.....	48
Fig. 42. Versão final do suporte das proteções laterais da mesa dedicada à robótica móvel.....	49
Fig. 43. Versão final do suporte já com a proteção lateral aplicada na mesa.	49
Fig. 44. Mecanismo de encaixe à pressão desenvolvido em Solidworks.	50
Fig. 45. Desenho CAD da peça embutida na base da mesa de trabalho.	51
Fig. 46. Desenho CAD da base da mesa de trabalho.	51
Fig. 47. Desenho CAD das placas laterais da mesa de trabalho.....	52
Fig. 48. Fase de acabamento da mesa de trabalho, à esquerda ainda no processo de passagem do tapa-poros sobre a superfície. À direita, a mesa após o processo de pintura e já com algumas placas do labirinto colocadas nos suportes.	52
Fig. 49. Corte a laser da placa MDF para a produção das paredes do labirinto.	53
Fig. 50. Balizas desenvolvidas para a atividade do Futebot (esquerda) e montagem dos para-choques no mBot2 (direita).	54
Fig. 51. CAD da garra desenvolvida (esquerda).e mBot2 no processo de manipulação de objetos (direita)	55
Fig. 52. mBot2 no fim do percurso do labirinto, destacado com fita vermelha na base (esquerda). mBot2 a percorrer a pista disponível no kit mBot2 (direita).	60
Fig. 53. Fotografia capturada durante o decorrer das atividades com os alunos da escola profissional.	61
Fig. 54. Luta de sumo entre dois robôs Lego Mindstorm, atividade realizada no âmbito da iniciativa Universidade de Inverno STEAM que decorreu na Universidade de Beira Interior.	61
Fig. 55. Display do modo “Dirigir” da aplicação Makeblock, à esquerda encontra-se o analógico que permite manipular a locomoção robô, os botões da direita realizam movimentos ou emitem sons predifinidos.	62
Fig. 56. Fotografias capturadas durante a realização do Futebot, torneio de futebol com os mBots. Destaque para as balizas e para-choques desenvolvidos para esta atividade recorrendo ao fabrico aditivo.	63

Lista de Tabelas

Tabela 1. Comparação dos resultados obtidos entre o grupo de controlo e o grupo experimental (amostra: 20 alunos), para as capacidade de resolução de problemas e criatividade. As pontuações foram estabelecidas segundo o método PSSS (<i>Problem-Solving Skill Scale</i>) e <i>Creative Thinking Abilities Test</i> (Çakır <i>et al.</i> , 2021).	13
Tabela 2. Protocolos desenvolvidos e respetivos recursos para a sua realização.	64
Tabela 2. Protocolos desenvolvidos e respetivos recursos para a sua realização (cont.).	65
Tabela 3. Percentagem de respostas afirmativas e negativas do primeiro questionário colocado aos participantes da formação dirigida ao ensino da robótica por via de kits de robótica móvel (mBot 2).....	68
Tabela 4. Percentagem de respostas afirmativas e negativas do último questionário colocado aos participantes da formação dirigida ao ensino da robótica por via de kits de robótica móvel (mBot 2).....	69

Nomenclatura

Acrónimos:

ANSI	<i>American National Standards Institute;</i>
CT	<i>Computational thinking (Pensamento computacional);</i>
ER	<i>Educational Robotics (Recursos robóticos educacionais);</i>
STEM	<i>Science, Technology, Engineering and Mechanics (Ciência, Tecnologia, Engenharia e Matemática);</i>
STEAM	<i>Science, Technology, Engineering, Arts and Mathematics (Ciência, Tecnologia, Engenharia, Artes e Matemática);</i>
JPA	<i>Japan Robotics Association;</i>
UNEC	<i>United Nations Economic Commission;</i>
IFR	<i>International Federation of Robotics;</i>
CS	<i>Computer Science (ciência computacional);</i>
K-12	<i>Kindergarten through grade 12 (jardim de infância até ao 12º ano de escolaridade);</i>
NRC	<i>National Research Council;</i>
UBI	Universidade da Beira Interior;
CSTA	<i>Computer Science Teachers Association;</i>
ISTE	<i>International Society for Technology in Education;</i>
AI	<i>Artificial Intelligence (Inteligência Artificial);</i>
IoT	<i>Internet of Things (Internet das coisas)</i>
AM	<i>Additive Manufacturing (Fabrico Aditivo)</i>
AR	<i>Augmented Reality (Realidade Aumentada)</i>
CPS	<i>Cyber-physical systems (Sistemas Ciberfísicos)</i>
CNC	<i>Computer Numerical Control (Controlo Numérico Computadorizado)</i>
CAD	<i>Computer Aided Design (Desenho assistido por computador)</i>

<i>FFF</i>	<i>Fused Filament Fabrication</i> (Fabricação por filamento fundido)
<i>CAM</i>	<i>Computer Aided Manufacturing</i> (Fabrico assistido por computador)
<i>PLC</i>	<i>Programmable logic controller</i> (Controlador Lógico Programável)
<i>RGB</i>	<i>Red, Green and Blue</i> (Vermelho, Verde e Azul)

1. Introdução

1.1. Enquadramento

O termo “robótica” surgiu pela primeira vez no conto “Runaround”, de Isaac Asimov, publicado em 1942. Leonardo Da Vinci projetou um cavaleiro que deveria mover-se automaticamente, mas a construção de um autómato com essas características só surgiu em 1962, por Joseph Engelberger e George Devol, com o nome de “Ultimate” (Diago *et al.*, 2022). A robótica define-se como o conjunto de técnicas tendentes a conceber sistemas capazes de substituírem o homem nas suas funções motoras, sensoriais e intelectuais (Javaid *et al.*, 2021).

A revolução industrial iniciou-se com a utilização de vapor para a produção de eletricidade, seguido da adoção da automação parcial nas linhas de montagem para produção em série, presentes na Indústria 2.0, até chegar à automação completa na Indústria 4.0. A Indústria 4.0 interligou avanços tecnológicos como Fabrico Aditivo, Inteligência Artificial, Realidade Aumentada, Sistemas Ciberfísicos, Sistemas Integrados, Robótica, IoT (*Internet of Things*), Análise de Dados, entre outros. Permitiu ainda que empresas e organizações conseguissem enfrentar desafios relacionados com a flutuação da procura e volatilidade do mercado. Também ajudou a melhorar a eficiência de produção. No entanto, a Indústria 4.0 enfrentou muitos desafios, como integração técnica, privacidade e segurança de dados, questões de gestão e a incapacidade dos trabalhadores em lidar com o aumento da digitalização (Khan *et al.*, 2023). Os funcionários necessitam de formação adequada para trabalhar num ambiente de Indústria 4.0 (Kiel *et al.*, 2017). Isso requer boas práticas de gestão para que estes trabalhadores possam lidar com as fábricas inteligentes (Shamim *et al.*, 2016). Estes problemas têm sido abordados pela filosofia aliada à Indústria 5.0, providenciando uma abordagem centrada no homem. Introduziu melhorias como manutenção preditiva, sistemas cognitivos ciberfísicos e robôs colaborativos. Porém, todas estas tecnologias requerem antes de mais a formação de técnicos especializados, que devem começar a adquirir conhecimentos e competências nos níveis mais baixos de escolaridade, i.e., o mais cedo possível, para que possam lidar adequadamente com tecnologias com um crescimento muito acentuado, tanto em termos de aplicabilidade como de complexidade.

Segundo o *World Robotics Report 2022* da *International Federation of Robotics*, a robótica encontra-se em expansão, com principal destaque a robótica industrial. Em 2021, foram instalados 517 385 robôs industriais em fábricas de todo o mundo. Representando uma taxa de crescimento de 31% de ano para ano e excede o valor máximo atingido antes da pandemia de instalação de robôs em 2018, que rondava os 11% (Como ilustrado na Fig. 1) Nos dias de hoje, o

stock de robôs operacionais a nível mundial atingiu um novo record de cerca de 3,5 milhões de unidades (Automation, 2022).

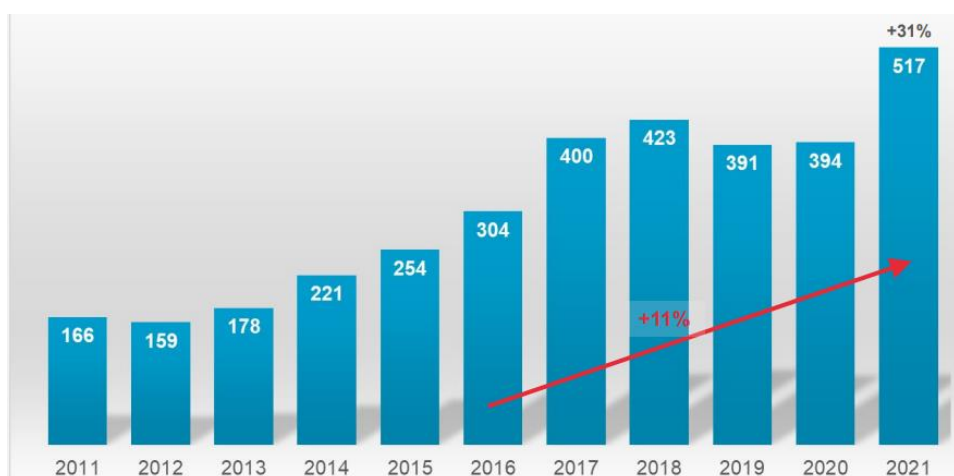


Fig. 1. Evolução da instalação de robôs industriais (em milhares de unidades) entre o ano de 2019 e 2021 (IFR, 2022).

As principais aplicações onde foram implementados, recentemente, robôs industriais são: manuseamento (*pick and place*); soldadura; montagem; limpeza; armazenamento (arrumação), processamento, etc. Estas aplicações, por sua vez, são essenciais quando se fala na indústria automóvel, alimentar, metal e maquinaria, entre outras (Fig. 2) (IFR, 2022).

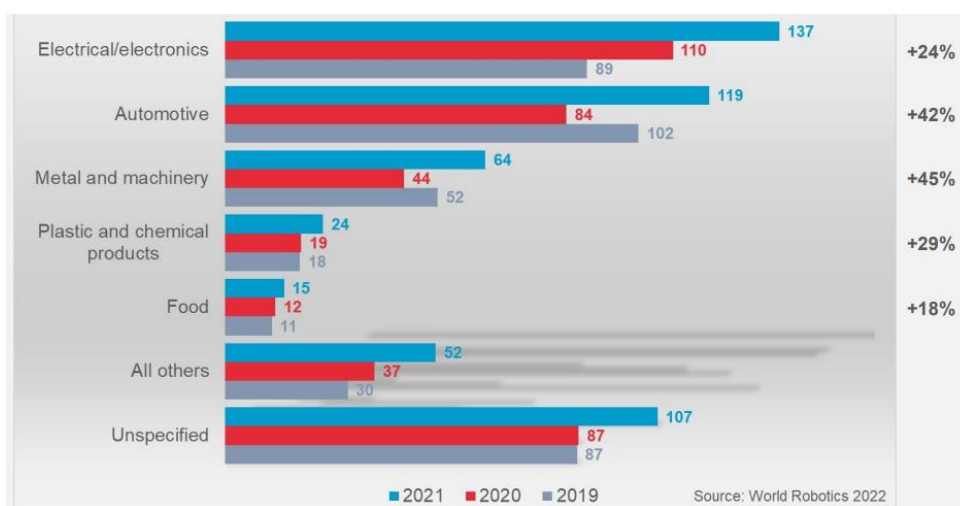


Fig. 2. Principais áreas da indústria onde ocorreu maior instalação de robôs industriais (em milhares de unidades) entre o ano de 2019 e 2021 (IRF, 2022).

O aumento do interesse verificado pela robótica ao nível da indústria traz consigo uma necessidade de preparar mão-de-obra qualificada para lidar com as inovações tecnológicas. Como

tal a implementação da robótica na educação é algo imperativo para o futuro da engenharia. A robótica pode ainda ser utilizada como um meio para as crianças desenvolverem o pensamento computacional (CT- *Computational Thinking*) através da utilização de recursos robóticos educacionais (ER- *Educational Robotics*) (Benitti, 2012). Existem diversas definições na literatura para o termo pensamento computacional. Wing (2006) define o termo pela primeira vez como sendo um processo iterativo, onde é aplicada abstração, dividindo um problema complexo em partes mais pequenas e através da razão heurística, encontrar uma solução. Esta abordagem não é apenas útil para problemas computacionais, mas também para a resolução de diversos problemas do dia-a-dia.

Os recursos educacionais robóticos são uma ferramenta poderosa e flexível, pois permitem que os estudantes descubram as coisas por eles próprios, aprendam novas linguagens de programação, desenvolvam metodologias de trabalho e pensamento crítico. Os alunos podem aplicar conceitos teóricos em projetos práticos de robótica, o que ajuda a solidificar o seu conhecimento e desenvolver habilidades técnicas, como programação, mecânica e eletrônica. Inúmeros investigadores argumentam que atividades que envolvam a programação de robôs aumentam o interesse dos participantes pelas áreas da ciência, tecnologia, engenharia e matemática (STEM- *Science, Technology, Engineering and Mathematics*) (Mead *et al.*, 2012).

1.2. O problema em estudo e a sua relevância

Na literatura existem diversos estudos sobre o efeito dos recursos robóticos nos alunos de diferentes níveis académicos (Atmatzidou & Demetriadis, 2016) (Çakır *et al.*, 2021) (Diago *et al.*, 2022). Diferentes autores destacam as vantagens e o potencial da robótica na educação, como ferramenta de aprendizagem, incluindo o apoio ao ensino de disciplinas que não estão necessariamente relacionadas com o campo da robótica (Benitti, 2012).

Num estudo realizado por Xia & Zhong (2018) foram analisados 22 artigos que descreviam estudos do tipo experimental, quasi-experimental e não-experimental com recursos educacionais robóticos em crianças do K-12. De uma forma geral as observações retiradas são positivas com os participantes a beneficiar de uma melhor compreensão de conceitos (programação de sensores, etc.), mudança de atitude (motivação, eficácia) e desenvolvimento de aptidões. Contudo Chambers *et al.* (2008) e Leonard *et al.* (2016) relatam nos respetivos trabalhos que apesar dos benefícios mencionados anteriormente, a maior parte dos estudantes não é capaz de providenciar explicações pormenorizadas sobre os conceitos mecânicos que sustentam a atividade realizada.

Apesar de todo o progresso realizado neste sentido, a aposta na implementação da robótica na educação ainda é encarada com alguma relutância, possivelmente pelos orçamentos limitados das escolas, a escassez de protocolos e metodologias que possibilitem a realização de atividades com robôs educacionais (ER). A falta de uma metodologia adequada pode conduzir à falta de interesse

e motivação por parte dos alunos. Outro aspeto importante é a evolução da complexidade destas atividades em função do nível académico em que as crianças se encontram (Xia & Zhong, 2018).

1.3. Objetivos e contribuição da dissertação

A dissertação tem como objetivo geral a implementação de uma abordagem STEM, utilizando kits de robótica móvel, pelo desenvolvimento de recursos educacionais integrados dirigidos a uma nova metodologia de ensino de robótica, com o intuito de motivar os estudantes a adquirir conhecimentos fundamentais para o futuro e promover o desenvolvimento da engenharia.

Para alcançar o objetivo geral, foram estabelecidos os seguintes objetivos específicos:

1. Desenvolvimento de protocolos laboratoriais para diferentes níveis académicos, que sejam sequenciais e permitam aprofundar os conhecimentos STEM (Ciência, Tecnologia, Engenharia e Mecânica), focados no ensino da robótica através de kits de robótica móvel;
2. Desenvolvimento do código e da infraestrutura de apoio necessária para a implementação dos novos protocolos laboratoriais;
3. Teste e validação dos protocolos desenvolvidos, garantindo a sua eficácia e adequação ao contexto educativo;
4. Recolha de dados para identificar oportunidades de melhoria nos protocolos laboratoriais, de forma a otimizar o processo de ensino e aprendizagem da robótica;
5. Discussão dos resultados obtidos e avaliação da evolução dos estudantes que foram submetidos à metodologia desenvolvida nos protocolos, analisando o impacto no seu conhecimento e habilidades relacionadas com a robótica.

A contribuição desta dissertação reside na criação de uma metodologia de ensino de robótica que seja progressiva, acessível a diferentes níveis académicos e estimulante para os estudantes. Além disso, a pesquisa irá fornecer perspetivas sobre o desenvolvimento de protocolos laboratoriais eficazes, que possam ser utilizados por outros formadores/professores e que contribuam para a promoção do interesse e do domínio da robótica no contexto educativo. Por fim, o desenvolvimento e construção de uma infraestrutura de apoio aos trabalhos com robótica móvel será uma contribuição que irá influenciar positivamente o ensino e aprendizagem em robótica.

1.4. Visão geral e organização da dissertação

Esta dissertação tem como objetivo principal explorar a utilização de robótica educativa no contexto da educação STEM (Ciência, Tecnologia, Engenharia e Matemática) e oferecer uma visão abrangente dos resultados obtidos. A estrutura da dissertação segue o seguinte esquema:

1. Introdução- Nesta secção, é apresentado o contexto e a importância da robótica educativa no ensino, destacando suas potenciais contribuições para o desenvolvimento de habilidades STEM. É discutido o problema de pesquisa que motivou este estudo,

delineando sua relevância no campo da robótica educativa e educação STEM. Os objetivos gerais e específicos desta dissertação são delineados, destacando a contribuição que esta traz para o campo da robótica educativa e educação STEM.

2. Estado da Arte- Neste capítulo, é apresentado um panorama do estado da arte da robótica educativa, com foco no pensamento computacional, estratégias e metodologias no ensino de programação/robótica, bem como uma revisão dos principais kits de robótica educativa utilizados. Também são apresentados os trabalhos realizados no âmbito da unidade curricular de Robótica Industrial
3. Materiais e Métodos- Esta seção descreve os materiais e métodos utilizados neste estudo, incluindo uma descrição detalhada dos robôs educacionais selecionados (mBot2 e Ultimate 2.0) e do *software* de programação mBlock 5. É descrita a construção da mesa de trabalho multiusos, a formação com mBot2 aos alunos de uma escola profissional, a participação na Universidade de Inverno STEAM (*Science, Technology, Engineering, Arts and Mathematics* - Ciência, Tecnologia, Engenharia, Artes e Matemática).
4. Resultados- Neste capítulo, são apresentados os resultados obtidos a partir da realização de questionários aplicados aos alunos que receberam esta formação, analisando as respostas dos participantes em relação à utilização da robótica educativa e sua influência no desenvolvimento de habilidades STEM.
5. Análise e Discussão de Resultados- Os resultados são analisados e discutidos em detalhes, relacionando-os aos objetivos propostos nesta dissertação e às contribuições para o campo da robótica educativa e educação STEM. São exploradas as implicações dos resultados obtidos e suas possíveis interpretações.
6. Conclusões- As conclusões gerais são apresentadas, resumindo os principais resultados e contribuições desta dissertação. São apresentadas sugestões de trabalhos futuros que possam expandir e aprofundar o conhecimento nesta área, explorando outras possibilidades e abordagens no campo da robótica educativa e educação STEM.

2. Estado da Arte

O interesse na robótica tem aumentado de forma abrupta nos últimos anos ao nível da indústria farmacêutica, na medicina, nos diversos setores da indústria, entre outras. A robótica é vista por muitos como uma forma de enriquecer a educação a todos os níveis (Johnson, 2003). O mercado de robôs educacionais também tem vindo a crescer, segundo a *Japan Robotics Association* (JPA), a *United Nations Economic Commission* (UNEC), e a *International Federation of Robotics* (IFR). De um ponto de vista pedagógico, as atividades com robôs educacionais na pré-escola e nos primeiros anos da escola primária são habitualmente baseadas em contacto físico e na relação sensor-motor. De acordo com autores neste campo, como Piaget, Bruner ou Fischbein, este tipo de atividades/experiências fornecem às crianças um envolvimento conceptual que começa com interações entre objetos do mundo real e desenvolve-se sofisticadamente através da descrição verbal (Lakoff & Núñez, 2000). Isto é, as crianças descrevem em voz alta todo o processo para resolver uma tarefa. Neste contexto, os robôs educacionais permitem que o conhecimento transmitido nas escolas não se fique por aspetos teóricos e seja possível a realização de atividades práticas que facilitam o processo de aprendizagem bem como o desenvolvimento cognitivo do pensamento computacional e outras competências necessárias para o século XXI (Bers *et al.*, 2013).

Nesta secção, são analisadas algumas abordagens e respetivos kits de robótica educativa utilizados, com o objetivo de identificar oportunidades de implementação de melhorias nas técnicas já existentes para o ensino da robótica. Através desta análise detalhada, são examinadas as características, funcionalidades e eficácia de diferentes abordagens e kits disponíveis no mercado. O objetivo reside em encontrar formas de aprimorar o processo de ensino e aprendizagem da robótica, considerando tanto as necessidades dos estudantes quanto as exigências do currículo educacional e do mercado de trabalho. Com base nessa pesquisa, foram selecionadas as melhores práticas e elementos-chave para a criação de uma metodologia inovadora e eficaz que estimule o interesse dos estudantes e promova o desenvolvimento de habilidades relevantes para a robótica e a engenharia.

2.1. Pensamento Computacional

Ideias que envolvem o pensamento computacional (*Computational Thinking* - CT) surgiram na década de 50 (Tedre & Denning, 2016). Papert (1980) foi o primeiro a descrever CT no seu trabalho relacionado com programação em Logo e a Logo *turtle* (Fig. 3), um robô educacional. No início dos anos 2000, o CT foi revitalizado por Wing (2006), uma vez que refinou a definição e realçou a importância que o CT tem como parte do conjunto de habilidades de qualquer criança. Definiu o CT como o processo de pensamento sistemático que os aprendizes utilizam enquanto

“resolvem problemas, projetam sistemas e compreendem o comportamento humano através do desenho de conceitos fundamentais para a ciência computacional (*Computer Science - CS*)”.



Fig. 3. Logo Turtle, o primeiro robô educacional desenvolvido por Seymour Papert (Solomon *et al.*, 2020)

Entretanto, ao nível da educação ainda não existe um consenso na definição de CT (Denning, 2017). Algumas definições de CT permanecem relacionadas com disciplinas na área da computação, nomeadamente a ciência computacional (Hamilton *et al.*, 2020). Outras definições foram criadas no contexto de outras unidades curriculares fora da CS. Por exemplo, Weintrop *et al.* (2016) realizou uma revisão da literatura sobre CT e entrevistou especialistas na área da matemática e ciência com o objetivo de desenvolver uma definição baseada em quatro categorias: ensaios de dados; ensaios de modelagem e simulação; ensaios de resolução de problemas computacionais; ensaios de *systems thinking*. Outros relacionam o pensamento computacional com a engenharia e existe ainda quem defina CT segundo uma abordagem multidisciplinar. Shute *et al.* (2017) afirma que CT é uma fundação conceptual necessária para resolver problemas de forma eficaz e eficiente.

2.1.1 Pensamento Computacional no K-12

A *National Research Council* (NRC) conduziu uma série de *workshops* com foco no CT, e consequentemente disponibilizou um relatório sobre as suas implicações educativas e cognitivas. O conjunto de participantes do *workshop* realizado pela NRC concordou que era necessário dar um próximo passo na realização de atividades semelhantes com maior foco nos aspetos pedagógicos do CT (National Research Council, 2010). Com o objetivo de implementar atividades de CT nas salas de aula do K-12, a *Computer Science Teachers Association* (CSTA) e a *International Society for Technology in Education* (ISTE) criaram uma equipa de líderes a nível

da educação e indústria para desenvolver uma estrutura capaz de integrar a ciência computacional e o pensamento computacional (CSTA Standards Task Force, 2011).

Diversas ferramentas foram utilizadas para ensinar estas componentes, algumas delas relacionadas com o trabalho desenvolvido por Papert ao nível da linguagem de programação educacional, logo, incluindo brinquedos educacionais e aplicações projetadas para crianças. Atualmente, é possível encontrar uma grande panóplia de kits robóticos no mercado (Hamilton *et al.*, 2020).

2.2. Estratégias e metodologias no ensino da programação/robótica

Existem diversas filosofias de ensino. A recomendação principal que surge da literatura é a de que o ensino se deve centrar para além da aprendizagem das características de determinada linguagem de programação, mas simultaneamente na combinação das mesmas e especialmente no problema adjacente referente ao projeto de programas básicos. Uma forma de o alcançar poderá ser através da introdução de muitos exemplos à medida que os programas são desenvolvidos, discutindo as estratégias usadas como parte deste processo (Robins *et al.*, 2003).

Segundo Coll *et al.* (2001), o professor deve: fazer a exposição dos conteúdos de forma gradual, com momentos de recapitulação, resumo e síntese; realizar analogias, usando os conhecimentos prévios dos alunos; ser elucidativo em relação às atividades propostas e ao que se quer ensinar, dar oportunidade ao aluno de executar os procedimentos de forma voluntária, consciente e inovadora e realizar aperfeiçoamentos. Os estudantes devem estar motivados para aprender os procedimentos e poderem autoavaliar-se, tendo o conhecimento de que a elaboração do conhecimento depende do seu esforço pessoal.

Os alunos devem desenvolver projetos concretos e reais, sendo necessário realizar algumas simplificações através de um método de desenvolvimento gradual (Pattis, 1990).

Há autores que realçam a importância da abordagem de tentativa e erro para que os alunos encontrem os erros de programação, utilizando a reflexão, compreensão, análise e teste de hipótese (Janzen & Saidan, 2006).

De acordo com Roumani (2006), o currículo devia ser lecionado de forma invertida, isto é, depois dos alunos estarem confortáveis com o comportamento e aplicações das principais estruturas de dados, é que devem aprender a implementá-las.

Os professores devem adotar estratégias que motivem o aluno a envolver-se na sua própria aprendizagem, e que lhe permitam desenvolver a autonomia. Na realização de desafios, deve privilegiar-se a crescente complexidade, apelando à articulação do conhecimento das várias

disciplinas, e que na resolução dos mesmos, os alunos cooperem em pequenos grupos (Rodrigues, 2020).

2.3. Robótica educacional

Como mencionado anteriormente, existe uma ampla variedade de robôs para todos os níveis de educação com diferentes finalidades (Hamilton *et al.*, 2020). Diversas investigações mostram que atividades com recurso a robôs educacionais conseguem ser bastante eficazes no desenvolvimento de capacidades como o pensamento crítico, criatividade, resolução de problemas, trabalho de equipa, tomada de decisão, entre outras (Benitti, 2012).

Geralmente a robótica tem sido aplicada na educação em perfis de estudantes com idade compreendida entre 3 e 18 anos, desde a pré-escola até a estudantes que frequentam o ensino secundário (Xia & Zhong, 2018). Segundo Xia & Zhong (2018), a maior fatia diz respeito a estudantes do ensino básico (57%), seguido de estudantes do ensino secundário (24%) e crianças do jardim de infância (19%). Mais de metade dos estudos realizados utilizam amostras com menos de 80 participantes e duração inferior a 2 meses. O tipo de robô dominante nos estudos foi o da marca LEGO (67%)

Tendo em conta a diversidade de oferta deste tipo de robôs educacionais, nesta secção vão ser abordadas algumas soluções disponíveis no mercado e estudos experimentais que abordam a influência do seu uso na aprendizagem.

2.3.1 Bee-Bot

O Bee-Bot é um dos mais conhecidos *floor robots* (“robô de chão”) no ensino básico e primário (Schina *et al.*, 2021). Com o formato de uma abelha, é categorizado como um “*Button-Operated Robot*” (Hamilton *et al.*, 2020), isto é, um robô comandado através de botões físicos situados na sua própria estrutura que podem ser pressionados em direções específicas. Habitualmente, a inclusão do Bee-bot nas atividades educacionais está associada com o desenvolvimento da capacidade de programar (Kazakoff *et al.*, 2013), capacidades cognitivas relacionadas com a resolução de problemas e flexibilidade cognitiva, noção espacial e pensamento algébrico através do planeamento de caminho. O Bee-Bot (Fig. 4 à esquerda) responde a comandos de movimento simples como virar à direita, virar à esquerda (realiza uma rotação de 90° sobre si próprio) seguir em frente ou recuar (realiza um movimento retilíneo de 15 cm para a frente ou para trás conforme o botão pressionado). Tem um botão de Pause, outro de Clear para remover todas as instruções, e por fim, um botão GO que executa a sequência de instruções (Diago *et al.*, 2022).

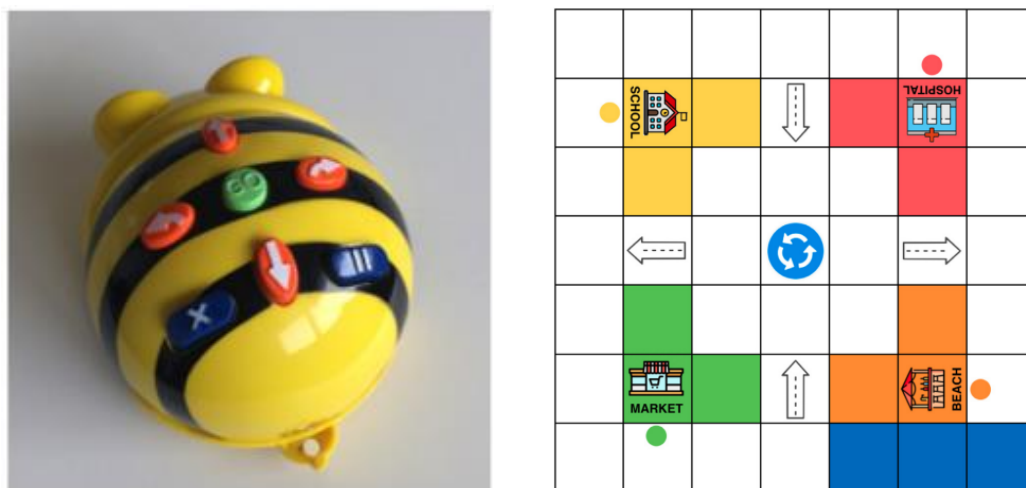


Fig. 4. Botões físicos do Bee-bot (esquerda) e o mapa utilizado na atividade experimental (direita) (Diago *et al.*, 2022).

Diago *et al.* (2022) compararam a execução de uma atividade seguindo instruções de ensino tradicionais e a realização dessa mesma atividade, mas com a utilização de recursos robóticos. Posteriormente foi analisado se os estudantes que realizaram a atividade com os recursos robóticos atingiram uma maior evolução no pensamento computacional e na noção espacial, do que os estudantes que seguiram as instruções de ensino tradicional. Antes da atividade, todas as crianças foram sujeitas à realização de um teste diagnóstico, igual para todas. A nível de pensamento computacional os resultados mostraram uma evolução superior nos alunos que utilizaram o Bee-Bot em relação ao grupo de controlo. Já em relação à noção espacial os resultados não são significativos, daí não ser possível estabelecer uma conclusão. A diferença significativa existente já nos pré-testes entre o grupo de controlo e o grupo experimental em relação ao pensamento computacional pode ser um fator limitante deste mesmo estudo.

Contudo, a utilização do Bee-bot revelou, de forma geral, um desenvolvimento cognitivo das crianças nas áreas avaliadas, reforçando assim o efeito positivo dos recursos robóticos na educação.

2.3.2 WeDo 2.0

Os kits de robótica WeDo 2.0 são conjuntos de peças e ferramentas projetadas para permitir que crianças construam e programem robôs simples. São desenvolvidos pela LEGO Education e incluem peças LEGO, motores, sensores e um *software* de programação fácil de usar chamado WeDo 2.0. Os kits são projetados para ensinar conceitos básicos de robótica e programação de uma maneira divertida e interativa. São frequentemente usados em salas de aula e em clubes de robótica para crianças e jovens (Çakır *et al.*, 2021).

O WeDo 2.0 foi utilizado numa investigação conduzida por Çakır *et al.* (2021), cujo objetivo era apurar o efeito dos recursos educacionais robóticos no desenvolvimento da capacidade de resolução de problemas, bem como o pensamento criativo das crianças na pré-escola. O procedimento experimental foi realizado ao longo de quatro semanas, um total de 32 horas. Durante este período, o grupo experimental realizou atividades com kits de robótica móvel WeDo 2.0, enquanto o grupo de controlo realizou atividades tradicionais (papel e caneta). Os participantes do grupo experimental utilizaram múltiplas configurações com diferentes graus de complexidade (de menor complexidade para maior complexidade) (Fig. 5-8). Em cada sessão era utilizado uma configuração diferente, todas elas programáveis de forma bastante intuitiva, através de um tablet ou computador (programação por blocos). O primeiro robô foi o Glowing Snail. Nessa atividade as crianças apenas controlavam qual a cor da luz emitida pelo robô (Fig. 5).



Fig. 5. Configuração Glowing Snail e respetiva programação.

O segundo robô foi o Cooling Fan. As crianças controlavam a potência de rotação do motor que fazia girar uma hélice (Fig. 6).

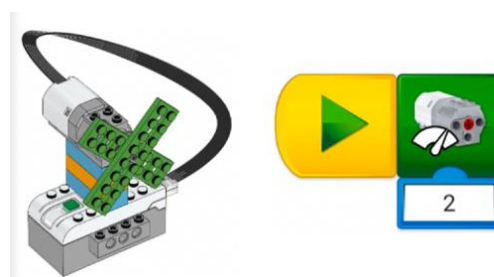


Fig. 6. Configuração Cooling Fan e respetiva programação.

O terceiro robô foi o MoovingSatellite. As crianças tinham a oportunidade de controlar a potência do motor, o sentido de rotação e a duração do funcionamento da mesma (Fig. 7).



Fig. 7. Configuração MoovingSatellite e respetiva programação.

O último robô foi o Frog Robot (Fig. 8). Com auxílio dos instrutores, as crianças montaram o robô e viram como se movia um sapo verdadeiro de forma a compreender melhor a complexidade do movimento. O robô apresentava um mecanismo simplificado e nesta atividade as crianças podiam controlar a potência do motor, o sentido da rotação, a duração do movimento e desligar o motor.



Fig. 8. Configuração Frog Robot e respetiva programação.

Tabela 1. Comparação dos resultados obtidos entre o grupo de controlo e o grupo experimental (amostra: 20 alunos), para as capacidade de resolução de problemas e criatividade. As pontuações foram estabelecidas segundo o método PSSS (*Problem-Solving Skill Scale*) e *Creative Thinking Abilities Test* (Çakır et al., 2021).

Capacidades avaliadas	Grupo	Média dos resultados antes do procedimento (0-72)	Média dos resultados depois do procedimento (0-72)
Criatividade	Experimental	13,40	22,90
	Controlo	13,85	16,90
Resolução de problemas	Experimental	31,2	43,10
	Controlo	29,6	35,20

Os resultados desta atividade permitem concluir que a robótica e a programação providenciaram um contributo significativo na capacidade de resolução de problemas em comparação às atividades do grupo de controlo. Posteriormente, os resultados dos pós-testes revelaram ainda melhorias significativas na imaginação e em fatores no domínio linguístico bem como no domínio do desenho.

2.3.1. Lego Mindstorms NXT

O kit Lego Mindstorms NXT (Fig. 9) é um dos mais populares e versáteis kits de robótica disponíveis no mercado. Inclui blocos de construção Lego, que lhes dá uma grande flexibilidade na criação de robôs de diferentes formas e tamanhos. O controlador programável NXT é o cérebro do robô, e pode ser utilizado para controlar os motores e adquirir os dados de sensores, bem como para armazenar e executar programas. O kit inclui três tipos de sensores: um sensor de toque, um sensor de cor e um sensor de distância ultrassônico. Os sensores permitem que o robô detete e atue em conformidade com o ambiente ao seu redor. Relativamente a atuadores, estão incluídos dois motores que permitem que o robô se mova. Os motores podem ser controlados para girar em velocidades e direções diferentes, permitindo que o robô se mova de forma precisa. O *software* de programação é fácil de usar, o algoritmo é construído por blocos de programação visual, semelhante a Scratch.



Fig. 9. Lego Minstorms NXT e o respetivo ambiente de programação Lego NXT (Atmatzidou & Demetriadis, 2016).

Existem muitas configurações possíveis desde robôs simples como seguidores de linha, até robôs mais complexos como robôs de navegação autónoma. Além disso, é possível instalar peças adicionais e modificações para criar robôs únicos (Atmatzidou & Demetriadis, 2016).

Num estudo realizado pela Universidade de Thessaloniki, Atmatzidou & Demetriadis (2016) investigam o desenvolvimento do pensamento computacional dos estudantes no contexto de atividades com robôs educacionais (ER – *Educational Robots*). O estudo aplica um modelo apropriado de pensamento computacional para explorar as capacidades dos alunos em dois grupos de idades diferentes (15 e 18 anos).

O trabalho foca-se ainda nos diferentes impactos possíveis que a abordagem de ensino pode ter no desenvolvimento do pensamento computacional dos estudantes em função da idade e do

género. Orientados através de fichas de trabalho, os alunos trabalharam em pequenos grupos para resolver problemas de programação.

O alvo do estudo foi um grupo de 164 estudantes de duas turmas diferentes, uma do 10^o ano de escolaridade e outra do 12^o ano. Todas as turmas tinham alunos do género feminino e masculino. Os níveis das capacidades de CT foram avaliados em diferentes fases da atividade, com foco em 5 parâmetros chave do CT- abstração, generalização, modelagem e decomposição.

O procedimento implementado dividiu-se em 8 seminários de treino com robôs. Cada seminário continha 11 sessões com a duração de 2h cada. Os seminários foram conduzidos por alunos pós-graduados (assistentes) e pelos professores das respetivas unidades curriculares em que eram realizadas as sessões.

Na primeira sessão, o professor introduziu a robótica no geral, o Lego Mindstorms NXT e o ambiente de programação Lego NXT. Foi realizado um questionário de perfil preenchido individualmente. Os estudantes, já reunidos em grupos, desenvolveram o seu primeiro programa usando um kit robótico. A ênfase foi colocada sobre o conceito de algoritmo e a importância de desenvolver instruções precisas que quando implementadas levavam à resolução de um problema.

Na segunda sessão, o objetivo passou por familiarizar os estudantes com conceitos de programação básicos como estruturas sequenciais e ciclos. Os estudantes foram também familiarizados com motores, sensores de toque, sensores de som e algumas características básicas do NXT, como a visualização de imagens a partir do ecrã do robô. Nesta sessão os estudantes programaram os robôs para dançar, sendo os principais conceitos abordados a abstração e a generalização.

Na terceira e na quarta sessão, os estudantes trabalharam na estrutura de controlo e como usar um sensor ultrassónico e foi introduzido o bloco de espera. Posteriormente, foi proposto o desafio de construir um programa que funcionasse como um sistema de alarme, capaz de detetar movimento e som. No fim da 4^a sessão os estudantes realizaram um novo questionário para avaliar o desenvolvimento da capacidade de CT.

Durante a 5^a e a 6^a sessão os estudantes exploraram a operação de um sensor de luz, criando um programa para construir um robô reciclador, capaz de seguir uma linha preta, e agarrar objetos, separando-os em recipientes, por cores.

Durante as restantes sessões o grau de dificuldade dos programas foi incrementado, acrescentando variáveis e operações aritméticas. Foi desenvolvido um programa em que o robô funcionava como um carro, seguindo as regras do trânsito. Foi realizado mais um questionário intermédio para avaliar o CT.

Na última sessão, o desafio final exigia que os grupos desenvolvessem um programa que resolvesse um determinado problema. O grupo que encontrasse a solução mais rápida e mais eficiente vencia o desafio.

Os resultados sugerem que os estudantes alcançam eventualmente o mesmo nível de desenvolvimento de habilidades de CT independentemente de sua idade e gênero. O CT, na maioria dos casos, necessita de tempo para se desenvolver completamente (as pontuações dos estudantes melhoram significativamente no final da atividade). As diferenças ao nível da criatividade, velocidade de execução e pensamento crítico relacionadas com a idade e gênero surgem ao analisar as pontuações dos estudantes nas diversas dimensões específicas do modelo de habilidades de CT. A modalidade do instrumento de avaliação de capacidades pode ter impacto no desempenho dos estudantes, isto é, a avaliação oral estimula melhor o pensamento crítico quando comparada com a avaliação escrita (Atmatzidou & Demetriadis, 2016).

2.3.2. mBot

O mBot (Fig. 10) é um robô educacional projetado pela Makeblock, direcionados para ciência computacional e aprendizagem STEM. A sua versatilidade e as inúmeras capacidades tornam-no numa solução interessante para ensino de robótica nas escolas e não só. Permite que os estudantes tenham a oportunidade de usufruir de aulas interativas, divertidas e com reflexão sobre aplicações das tecnologias de ponta no mundo real (Pisarov & Mester, 2019). Na sua constituição base está um processador CyberPi, compatível com as linguagens de programação Python e Scratch (programação por blocos). Este robô contém um sensor ultrassónico, um sensor seguidor de linha e dois motores com *encoder*. O chassis permite ainda a implementação de um conjunto de *add-ons* como sensores de temperatura, sensores de gás, acelerómetro, giroscópios, entre outros.

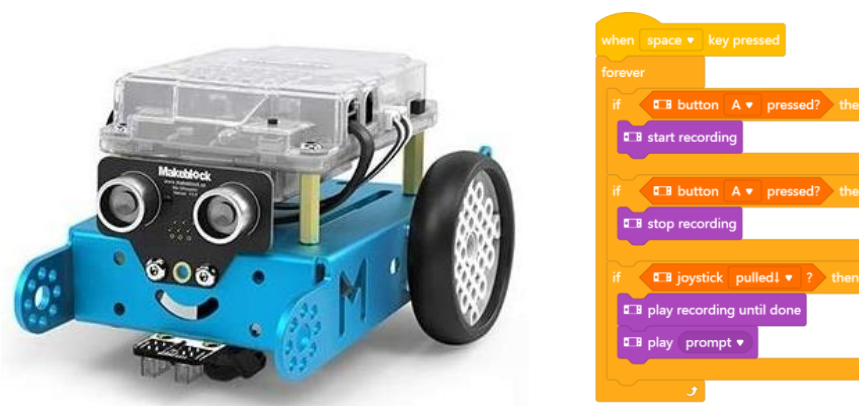


Fig. 10. mBot e o respetivo ambiente de programação Scratch.

Na universidade *Matej Bel*, na Eslováquia, Voštinár (2020) realizou um conjunto de atividades e *workshops* utilizando mBots, com o objetivo de motivar os estudantes do ensino básico e secundário a aprender programação. Um dos *workshops* teve como alvo de estudo alunos do 7º e 8º com idades compreendidas entre os 13 e 14 anos. A primeira preocupação foi perceber se o grupo tinha alguma experiência com robôs educacionais. Todos eles já tinham usado o Lego Mindstorms e se tinham noções básicas de Scratch. Com base nos dados recolhidos, foram lançados 3 desafios cujo grau de complexidade aumentou progressivamente. No primeiro desafio, apenas foi pedido que os estudantes desenvolvessem um programa que simulasse as luzes de um carro (o vermelho representava as luzes de travagem, e o laranja era emitido na realização de curvas tanto à esquerda como à direita, seguido de um sinal sonoro para simular a buzina).

A segunda tarefa já exigiu que a velocidade de locomoção do mBot aumentasse de forma progressiva, através de uma sequência de blocos que incrementou a velocidade em intervalos de 20% a cada 0,5 segundos. No terceiro desafio, foi exigido que a sequência de blocos fosse substituída por um ciclo no qual a cada 0,5 segundos, a variável que define a velocidade aumentasse 20%. Desta forma foi possível familiarizar os estudantes com o conceito de ciclo e contadores.

No final das atividades realizaram-se questionários como o objetivo de medir o interesse dos participantes e perceber o grau de dificuldade que os alunos sentiram durante a resolução dos problemas. No geral foi recebido um feedback positivo por parte de quase todos os estudantes.

2.3.3. Ultimate 1.0

O Ultimate 1.0 (Fig. 11) é um dos muitos kits que a Makeblock tem atualmente disponíveis no mercado, caracterizados pelo uso de fortes estruturas de alumínio, bem como a montagem clássica através de componentes mecânicos como parafusos e porcas.

A diversidade de peças do Ultimate Robot Kit permite construir uma grande variedade de robôs, que se podem tornar bastante complexos. Entre diversos componentes que estão incluídos no kit, existem três motores DC com caixa redutora (*gearbox*), uma garra cuja abertura/fecho é controlada através de um motor DC, lagartas, um sensor ultrassónico, um sensor infravermelho capaz de detetar linhas, um módulo de comunicação Bluetooth, LED RGB, e um controlador Orion. Uma das grandes vantagens deste controlador é o facto de ser compatível com Arduino, permitir que a sua programação seja realizada através do MBlock, isto é, uma programação gráfica baseada em Scratch e ainda através de Python (Candelas *et al.*, 2016).



Fig. 11. Algumas configurações possíveis de montagem com as peças contidas no kit Ultimate 1.0.

Candelas *et al.* (2016) realizaram uma atividade com estudantes do curso de Engenharia Robótica da Universidade de Alicante. O principal objetivo foi desenvolver uma configuração capaz de percorrer um trajeto com obstáculos.

Os alunos foram divididos em grupos de 4 a 5 elementos e a metodologia seguida pelos investigadores dividia a interação com os alunos em 5 atividades.

Na primeira atividade foi introduzido o conceito de projeto, como planejar e documentar, definir objetivos e prazos para os mesmos.

Durante a segunda atividade foi introduzido o Arduíno IDE (*Integrated Development Environment*) uma vez que o controlador Orion pode ser programado utilizando esta plataforma. Apesar do Scratch ser uma linguagem muito intuitiva, o Arduíno IDE foi a plataforma escolhida, por permitir atingir uma maior complexidade, adequando-se a uma amostra de estudantes universitários. A linguagem utilizada para programar no Arduíno IDE é uma variação do C++, que se adequa melhor, quando comparada com o Scratch, ao objetivo de preparar os alunos para o mercado de trabalho.

Após a introdução de conceitos fundamentais para a programação no Arduíno IDE, como funções “*setup*”, “*loop*” e a criação de funções que controlam atuadores em função da informação lida por um sensor, foi proposto que os estudantes desenvolvessem um programa capaz de controlar dois

motores DC, inicialmente em função do intervalo de tempo, e posteriormente de acordo com a distância medida por um sensor ultrassônico.

Na terceira atividade foi introduzido o kit robótico Ultimate, e os estudantes tiveram a oportunidade de explorar todas as peças disponíveis e desenvolver a capacidade necessária para as montar. Após a montagem, cada grupo teve a oportunidade de controlar o robô por *Bluetooth* através de uma aplicação disponível para *Smartphones*. Finalmente, foi lançado o desafio de criar um programa que permitisse ao robô mover-se de forma autônoma, evitando obstáculos.

A quarta atividade foi dividida em várias sessões, onde se desenvolveu o projeto do robô capaz de percorrer um trajeto, contornando todos os obstáculos e realizando algumas tarefas durante o percurso. Nesta fase os alunos precisaram de pensar nas alterações necessárias para alcançar o objetivo.

A Fig. 12 ilustra o cenário proposto pelos investigadores que foi previamente exibido aos alunos antes da fase do projeto se iniciar.

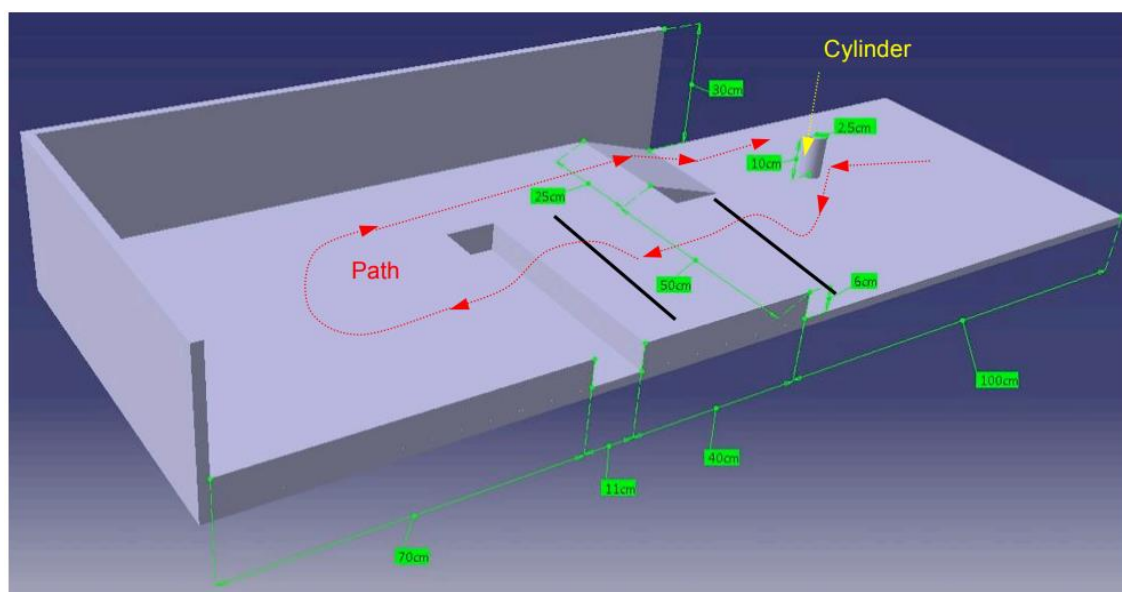


Fig. 12. Esquema e dimensões do circuito teste proposto pelos investigadores.

A última atividade consistiu na aplicação do projeto, através de uma competição entre grupos com o objetivo de demonstrar qual a configuração e respetiva programação do Ultimate com maior capacidade de superar as barreiras impostas pelo cenário no menor tempo possível. Para além do

tempo, existiram alguns fatores que determinaram a pontuação de cada grupo: a originalidade do design; a correta distribuição de componentes; o uso de diferentes sensores e atuadores; a forma como o robô realiza uma determinada tarefa e a estrutura do programa.

Em todas as sessões os grupos explicaram o programa desenvolvido e justificaram as decisões tomadas na realização do mesmo.

De uma forma geral, foi possível concluir que os estudantes mostraram um grande interesse e foco na resolução dos problemas que surgiram ao longo das atividades. O facto de a plataforma Arduino IDE ter um ambiente e uma comunidade que facilita a aprendizagem rápida, também contribuiu positivamente para os resultados. Os tipos de atividades desenvolvidas adequam-se a uma unidade curricular introdutória ao tema da robótica, permitindo uma compreensão da função e aplicação de sensores, atuadores, a lógica de programação a partir de um contacto prático com sistemas robotizados.

2.4. Trabalhos realizados no âmbito da unidade curricular de Robótica Industrial- Primeiro contacto

A robótica educativa tem ganho cada vez mais destaque como uma ferramenta eficaz no ensino de conceitos teóricos e práticos relacionados à robótica e à automação. No âmbito das unidades curriculares de Robótica, Sistemas Robotizados e Robótica Industrial da Universidade da Beira Interior, os alunos são desafiados a aplicar o raciocínio e os métodos adquiridos teoricamente durante o período de aulas por meio da realização de trabalhos práticos. Essas atividades visam proporcionar aos estudantes a oportunidade de colocar em prática os conhecimentos adquiridos e de se familiarizarem com as ferramentas e técnicas atualmente utilizadas na indústria.

Ao longo do semestre da unidade curricular de Robótica Industrial, foram abordados diversos tópicos que visam munir os estudantes com as habilidades necessárias para enfrentar os desafios da indústria robótica. Para viabilizar a aplicação prática desses conhecimentos, três plataformas robóticas foram disponibilizadas: mBot 2, Ultimate 2.0 e Petoí Bittle. Cada grupo de alunos selecionou uma dessas plataformas e teve como objetivo concluir um desafio proposto.

Neste subcapítulo, serão abordados três trabalhos práticos que se destacam por apresentarem um nível de complexidade mais elevado, o que se adequa ao ambiente universitário e à formação avançada dos alunos. O primeiro trabalho consiste na construção de uma torre de blocos de madeira utilizando o kit Ultimate 2.0. Os alunos enfrentam o desafio de projetar e programar um robô capaz de manipular os blocos e construir uma estrutura estável.

No segundo trabalho, os estudantes são desafiados a desenvolver um robô capaz de praticar tiro ao alvo, também utilizando o kit Ultimate 2.0.

Por fim, no último trabalho abordado, os alunos têm o desafio de programar um mBot2 para resolver um labirinto.

Esses três trabalhos práticos representam exemplos concretos de como a robótica educativa pode proporcionar aos estudantes uma experiência enriquecedora na aprendizagem de conceitos fundamentais da robótica e automação. Ao enfrentarem desafios complexos e trabalharem em equipa, os alunos têm a oportunidade de aplicar os seus conhecimentos teóricos de forma criativa e inovadora, preparando-se para enfrentar os futuros desafios da robótica industrial.

2.4.1. Construção de uma torre de blocos utilizando o Ultimate 2.0

A configuração escolhida do Ultimate 2.0 para a construção da torre de peças foi a “robotic arm tank” (Fig. 13), não só por ser a única das 10 configurações base dadas pelo produtor que permite facilmente manusear objetos por conter uma garra na ponta de um braço robótico, como também por ser a configuração que necessita de menos alterações e/ou blocos/extensões adicionais para o objetivo pretendido.

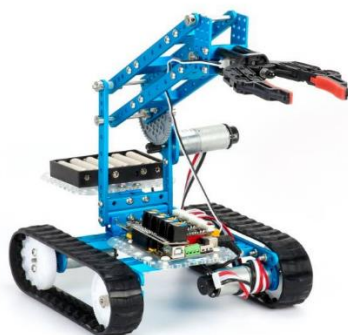


Fig. 13. Configuração “robotic arm tank”, uma das configurações base do kit Ultimate 2.0

Como se pode observar na Fig. 13, esta configuração utiliza 2 lagartas controladas, cada uma pelo respetivo motor, para mover o robô. Contém fixo à sua base um braço que suporta, posteriormente, um antebraço que pode ser inclinado, através de um 3º motor, para ajustar a altura da garra que se encontra na sua extremidade. Devido à sua configuração de montagem, a garra permanece sempre na posição horizontal, independentemente da inclinação do antebraço.

A garra, como dito anteriormente, contém um motor DC, que a permite abrir e fechar, permitindo a manipulação de objetos.

Foram efetuadas modificações no hardware do robô. Primeiramente, não existia nenhum sensor que permitisse a deslocação controlada do robô. Com esse fim, foi instalado um sensor infravermelho na parte inferior do chassi. Este sensor é composto por dois pares emissor/recetor. Após a conexão do sensor ao controlador MegaPi através de uma placa com conexão RJ25, o output recebido pelo controlador varia entre 0 e 3. Sendo que 0 (zero) corresponde à situação em que ambos os recetores identificam a linha preta, ou seja, o feixe infravermelho emitido não é refletido pela superfície até ao recetor. Na ocasião em que apenas o recetor da direita é acionado, o output é 1; se apenas o recetor da esquerda for acionado, então o output será 2. Finalmente, se ambos os recetores forem acionados, o output é 3, não sendo detetada linha.

O sensor foi posicionado o mais à frente possível, permitindo a antecipação da linha e uma correção eficaz da trajetória. A posição deste sensor pode ser observada na Fig. 14. Efetuada a instalação do sensor de linha e otimização do algoritmo, o grupo de alunos presenciou a falta de força nos motores para o robô virar 90° sobre si. Todavia, o kit Ultimate 2.0 contém várias peças mecânicas que permitiram um aumento da largura do chassi, afastando as lagartas do centro de rotação. Desta forma observou-se um maior momento devido ao acréscimo da distância sobre a qual a força de viragem é aplicada.

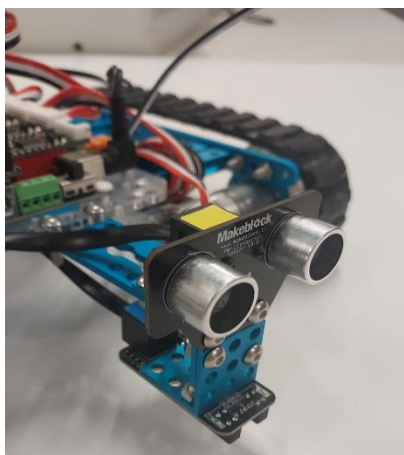


Fig. 14. Montagem do sensor ultrasónico e do sensor infravermelho.

Seguindo o raciocínio de funcionamento, a próxima limitação do kit original seria a possibilidade de identificar os blocos de madeira. O sensor ultrasónico incluído no kit foi utilizado para esse fim. Este sensor foi montado na parte da frente do “robotic arm tank”. Desta forma é possível

determinar não só a presença de um bloco na sua trajetória, mas também a que distância se encontra.

Uma vez que o robô consegue identificar a presença e a distância a que se encontram os blocos, foram efetuados vários testes com a finalidade de averiguar a distância perfeita entre o sensor ultrassônico e o bloco, para que a garra ao baixar pudesse apanhar o mesmo. No entanto, a configuração presente na frente do robô provocaria uma colisão do antebraço com o sensor ultrassônico. Para resolver este conflito, a garra foi reinstalada numa posição superior como pode ser observado na Fig. 15.

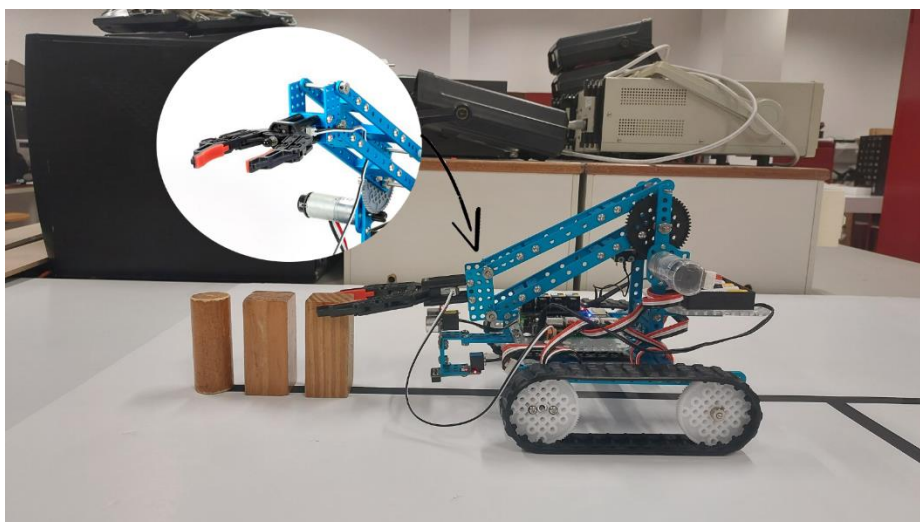


Fig. 15. Comparação entre a posição original da garra e a posição utilizada.

A última modificação efetuada foi a adição de um interruptor fim de curso no braço robótico. Esta adição surgiu da necessidade de impedir o choque do braço sobre a placa. O interruptor (Fig. 16) permite ao processador registar quando é que o braço está prestes a colidir e com base nessa informação, cancelar o movimento descendente.

Antes de realizar o fluxograma, que por sua vez, daria origem ao programa, foi necessário estabelecer um caminho/trajeto. A configuração escolhida pelo grupo assemelha-se a um “T” (Fig. 17).

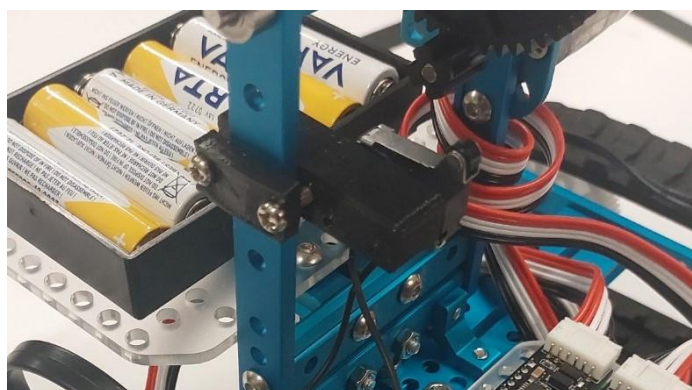


Fig. 16. Localização do sensor fim de curso no braço robótico que previne a colisão com a placa de controlo.

Foi posteriormente estabelecida a localização inicial do robô, a localização dos blocos e a localização da construção da torre de blocos definidos na Fig. 17 como Robô, Blocos e Torre, respetivamente.

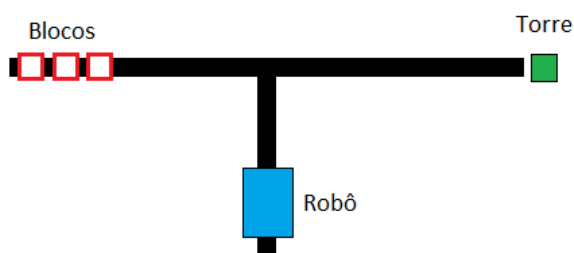


Fig. 17. Localização inicial do Robô e dos Blocos, com a localização onde serão empilhados os blocos

2.4.2. Tiro ao alvo com o Ultimate 2.0

Para este trabalho, o objetivo de lançar um objeto (flecha), exigiu a construção de uma configuração totalmente distinta das configurações originais apresentadas pelos criadores do kit Ultimate 2.0.

Todo o desenvolvimento estrutural deste robô foi possível graças a processos de fabrico aditivo.

Estruturalmente, partindo do módulo apresentado na Fig. 13, desenvolveu-se a estrutura apresentada na Fig. 18. Destacando como principal diferença, a exclusão da elevação da garra, uma vez que não se adequava ao objetivo pretendido.

Optou-se pelo desenvolvimento de um sistema mecânico de pinhão e cremalheira, projetados com auxílio do software de Desenho Assistido por Computador (CAD – *Computer Aided Design*) Solidworks. Tanto a cremalheira, como os seus suportes, o alvo e as flechas foram produzidos numa Impressora 3D. O material utilizado foi PLA devido à sua baixa probabilidade de deformação, rigidez elevada o suficiente para os requisitos do projeto e fácil impressão. O arco foi impresso em PETG devido à superior resistência à flexão do material.

A implementação do sistema pinhão cremalheira partiu de um pinhão já existente no kit. Foram retiradas as medidas do diâmetro e número de dentes de forma a calcular o módulo da roda dentada. Posteriormente construiu-se uma cremalheira compatível com o pinhão.

No final da cremalheira foi desenvolvida uma saliência com função de travar o movimento em excesso da mesma. Nesta peça, foi acrescentado um apoio para fixar a garra. Todos os componentes desenvolvidos estão destacados na Fig. 19.

O desenvolvimento deste sistema visa estabelecer um movimento linear da garra até ao arco, e em seguida puxar a flecha.

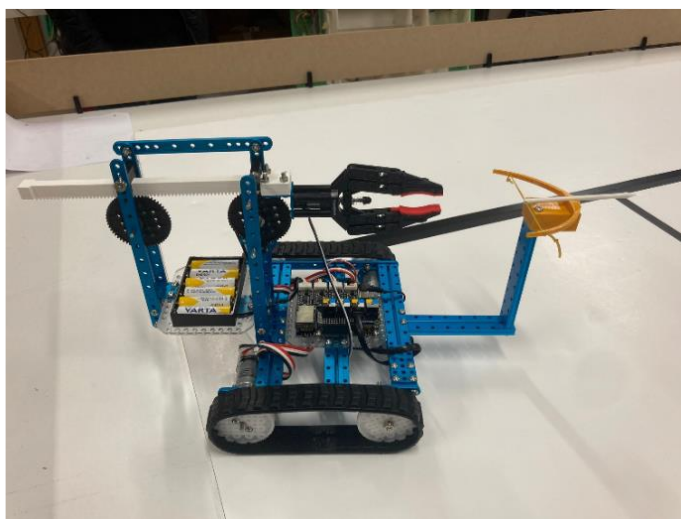


Fig. 18. Configuração construída com componentes do kit Ultimate 2.0 e componentes desenvolvidos através da fabricação aditiva.

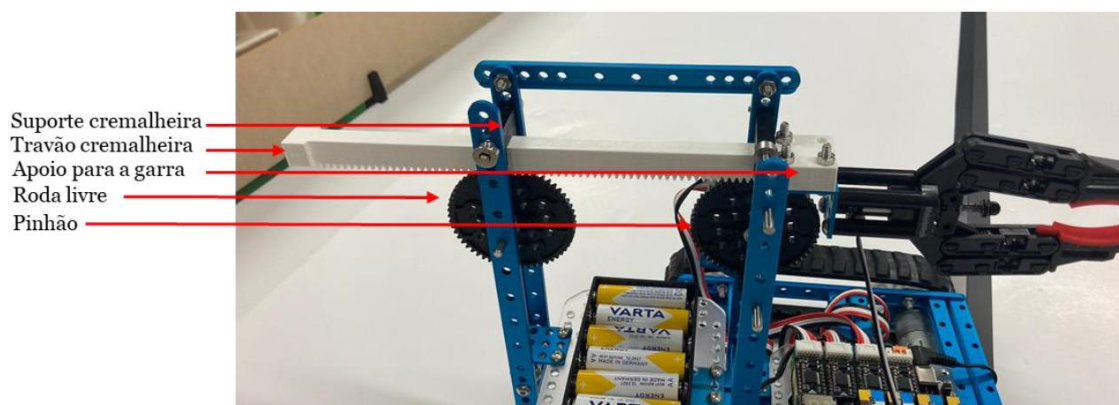


Fig. 19. Identificação dos componentes no sistema pinhão cremalheira.

Relativamente ao mecanismo de arco e flecha, alguns elementos foram considerados até a versão final ter sido projetada. Em primeiro lugar, o arco possui um suporte que lhe confere a altura e a estrutura necessária para lançar flechas. Na Fig. 20, é possível observar uma ranhura criada para a acomodação da flecha.

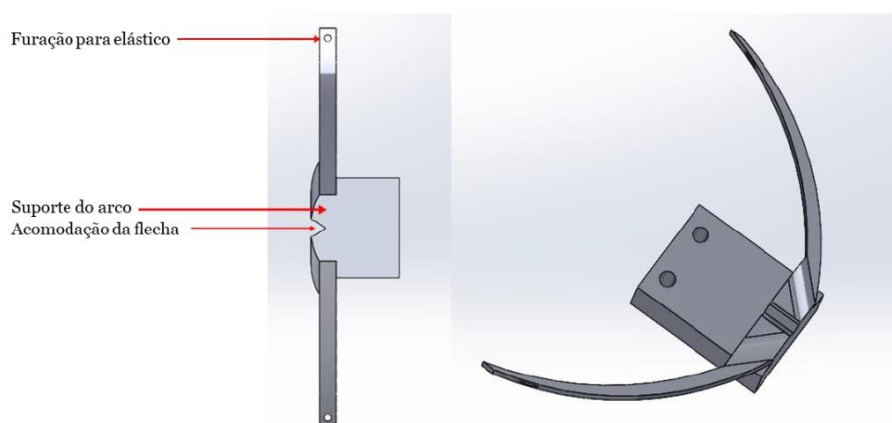


Fig. 20. Arco desenhado em Solidworks implementado na configuração final do robô.

À semelhança do que aconteceu com o arco, várias versões da flecha foram criadas. Na Fig. 21, pode-se constatar algumas das possibilidades exploradas. Após alguns testes, concluiu-se que a flecha com quatro alhetas revelava ser a que melhor se comporta em termos de lançamento.

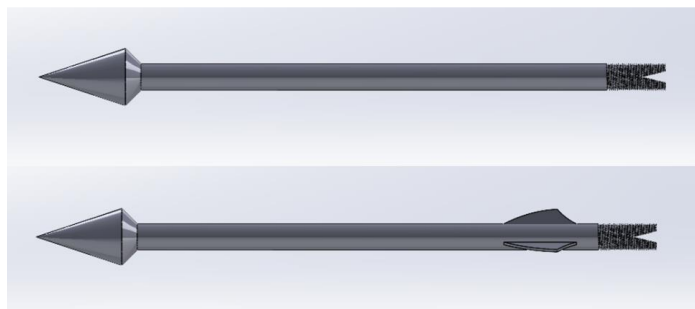


Fig. 21. Exemplos de versões de flechas desenhadas em Solidworks.

Para facilitar a tração entre a garra e a ponta traseira da flecha, foi desenhada uma textura que permitiu aumentar o atrito entre esta zona da flecha e os dedos da garra. A fissura visível nesta extremidade é onde assenta o elástico.

Por fim, o desenho do alvo foi retirado da plataforma Thingiverse. A escolha deste alvo deve-se ao facto de o setor atingido cair, sendo possível detetar qual a zona atingida pela flecha.

Na Fig. 22 encontra-se o alvo utilizado no projeto, juntamente com o conjunto arco e flecha.



Fig. 22. Alvo utilizado no projeto tiro ao alvo (esquerda), conjunto arco e flecha desenvolvido em Solidworks (direita).

Para definir o itinerário, aplicou-se fita adesiva de cor preta no piso, obtendo um segmento de reta. Através de um sensor infravermelho incorporado na parte inferior do robô, foi realizada a correção da trajetória e no final do percurso, o robô lançou a flecha ao alvo. Após o lançamento, o robô retorna à sua posição base ficando assim apto a realizar um novo ciclo.

2.4.3. Resolução de labirinto com mBot2

A planificação deste projeto tinha como requisito obrigatório colocar o robô a percorrer um labirinto aleatório, sendo necessária uma estrutura que permitisse criar vários percursos.

Para a construção da base do cenário, o grupo de alunos utilizou 2 placas MDF lacadas (3 mm de espessura) unidas com fita adesiva. Assim sendo, o labirinto (Fig. 23, à esquerda) possui uma dimensão de 1220 x 1220 mm. As paredes do labirinto também são placas MDF (3 mm de espessura), no total foram cortadas 41 placas com 21,5 x 12 cm. Para suportar as placas, foram desenhadas colunas com 100 mm de altura com ranhuras na sua configuração que permitem o encaixe das paredes internas do labirinto (Fig. 23, à direita). Todas as colunas foram coladas à base do labirinto.

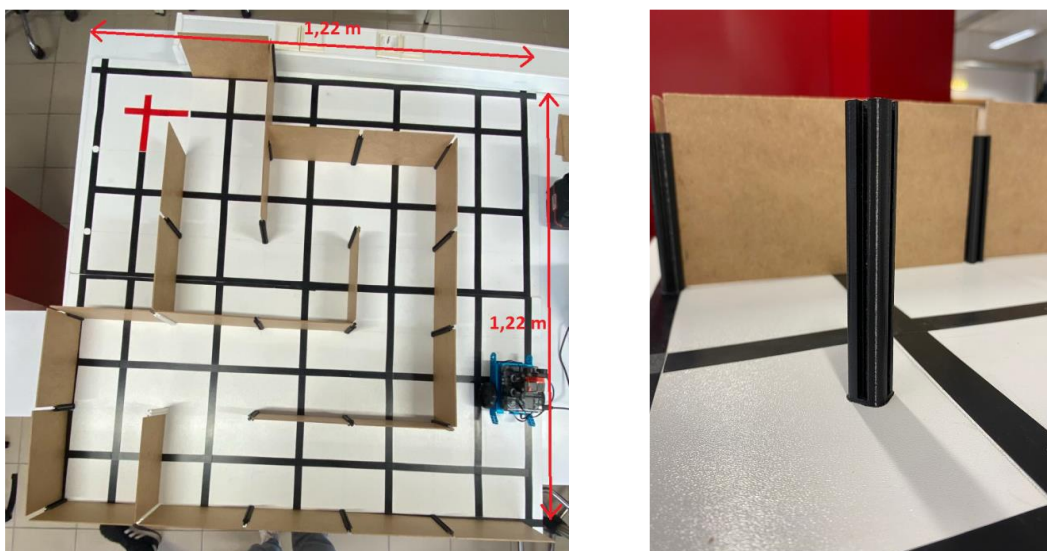


Fig. 23. Labirinto projetado com as respetivas dimensões exteriores (esquerda), coluna que suporta as paredes do labirinto, desenhada em Solidworks, produzida recorrendo fabrico aditivo (direita).

Na base do labirinto foi colocada fita adesiva preta com o intuito de formar uma grelha que auxilia o robô na resolução do labirinto, permitindo que se mantenha o mais paralelo possível com as placas que limitam o percurso. A correção da trajetória é realizada com auxílio do sensor RGB disponível no kit original do mBot2.

Para facilitar a resolução do labirinto, o grupo de alunos realizou algumas alterações na configuração do mBot2. Foi projetada uma peça que permite a implementação de mais um sensor ultrassom no robô, como ilustra a Fig. 24.

Desta forma, foi possível estabelecer uma estratégia de tomada de decisão em função da existência de paredes à frente e/ou à esquerda do robô.

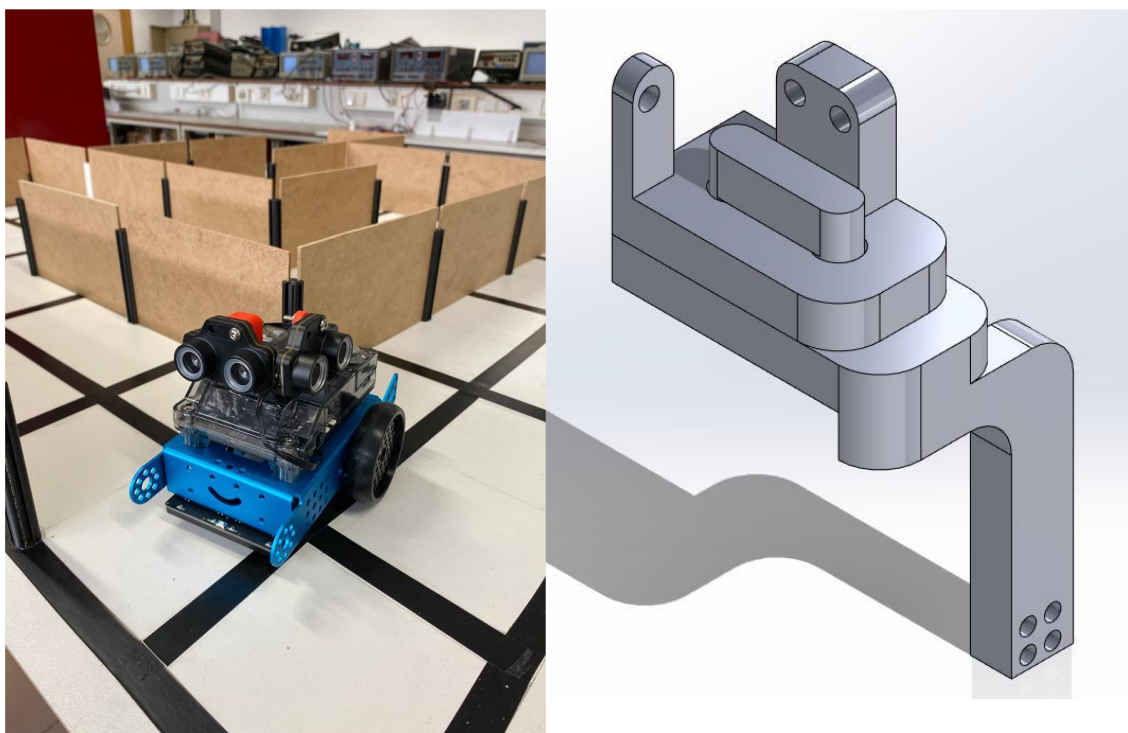


Fig. 24. mBot2 com dois sensores de ultrassons implementados (esquerda), desenho CAD do suporte para os sensores ultrassônicos (direita).

Outro fator que levou à criação da peça presente na Fig. 24 (à direita), foi o facto da distância mínima medida pelos sensores ultrassónicos ser de 4 cm, não permitindo uma leitura correta quando estes se encontravam muito próximos da parede, levando o robô a colidir com os limites do labirinto.

Com o design apresentado na Fig. 24, foi possível montar os sensores numa posição mais próxima do centro do mBot2 e estabelecer uma distância superior às paredes do labirinto, facilitando a leitura dos sensores.

2.5. Nota conclusiva

Nesta secção, foi conduzido um estudo abrangente de diversas tecnologias e metodologias aplicadas ao ensino da robótica. Foram abordados alguns trabalhos no âmbito das unidades curriculares de Robótica, Sistemas Robotizados e Robótica Industrial da Universidade da Beira Interior, onde se estabeleceu o primeiro contacto com os kits de robótica móvel mBot2 e o

Ultimate 2.0. Com base no conhecimento existente, o objetivo principal desta dissertação é contribuir para a expansão do conhecimento no campo da robótica educacional. Através de uma abordagem cuidadosa e ponderada, pretende-se introduzir abordagens e metodologias inovadoras que promovam efetivamente o envolvimento dos estudantes na engenharia. Ao abordar as lacunas presentes nas estratégias atuais, busca-se aprimorar a eficácia global da educação na robótica, preparando os estudantes para os desafios e oportunidades em constante crescimento nesse campo em desenvolvimento.

3. Materiais e Métodos

3.1. Robôs educacionais

Os robôs educacionais são programados pelo seu utilizador para agir segundo uma determinada informação proveniente do ambiente onde se encontra inserido. Têm na sua constituição um conjunto de sensores, que permitem medir vários tipos de condicionantes e transmitir essa informação ao controlador do robô. Existe um extenso espetro de sensores, desde sensores de luz, de toque, de temperatura, de humidade, de rotação, de som, de cor, de distância. Simultaneamente, o robô possui atuadores, que, como o próprio nome indica, permitem atuar sobre o ambiente onde está inserido. Habitualmente motores, que possibilitam a adição de vários mecanismos, como braços robóticos, rodas, sistemas de transmissão (*gearbox*).

Após uma análise cuidada de alguns artigos e estudos, verifica-se que existe uma vasta oferta de robôs educativos. Algumas opções permitem até construir robôs usando materiais de baixo custo ou recicláveis. Associado ao crescimento desta área da robótica, têm surgido simultaneamente diversos ambientes de programação por blocos, projetados para serem utilizados por crianças, permitindo assim a programação e a interação, facilitando os primeiros contactos com a área da programação, além de contribuir para a formação escolar.

Para a realização dos protocolos e das respetivas atividades associadas aos mesmos, optou-se por uma versão mais recente dos robôs previamente mencionados no capítulo anterior. Nesta secção serão abordados todos os aspetos técnicos dos kits de robótica móvel utilizados no envelope de trabalhos da dissertação, o mBot2 e o Ultimate 2.0, da Makeblock, assim como o ambiente de programação, mBlock.

3.1.1. mBot2

O mBot2 (Fig. 25) é um robô educacional projetado numa abordagem STEAM (Makeblock, 2022). As suas extensas capacidades tornam-no uma boa opção para a introdução da robótica para os níveis académicos primários, contudo pode ser explorado numa vertente de ensino secundário e até universitário.

O microcontrolador utilizado no robô é o CyberPi, comandado através do *software* mBlock, compatível com linguagens de programação como Scratch e Python. O CyberPi apresenta uma

grande versatilidade, pois apresenta na sua constituição um conjunto de sensores e atuadores como microfone, colunas, uma unidade de medição de inércia com giroscópio e acelerómetro, um sensor de luz, botões de operação (incluindo um *joystick*) e um ecrã colorido. Complementado o CyberPi, o mBot traz no seu kit um sensor ultrassónico e um sensor seguidor de linha com 4 elementos RGB. Para além de toda esta gama de sensores e atuadores, o mBot2 permite a comunicação através de Wi-Fi. Os mBots podem ser conectados de forma a criar uma rede local onde os robôs podem comunicar entre si, sem qualquer necessidade de cabelagem (*wireless*).

O mBot2 pode ser ligado à Internet e desempenhar funções como reconhecimento de voz ou aceder a bibliotecas que contêm funcionalidades diversas como *Machine Learning*. Na Fig. 25 é possível observar os diferentes componentes na constituição do mBot2.

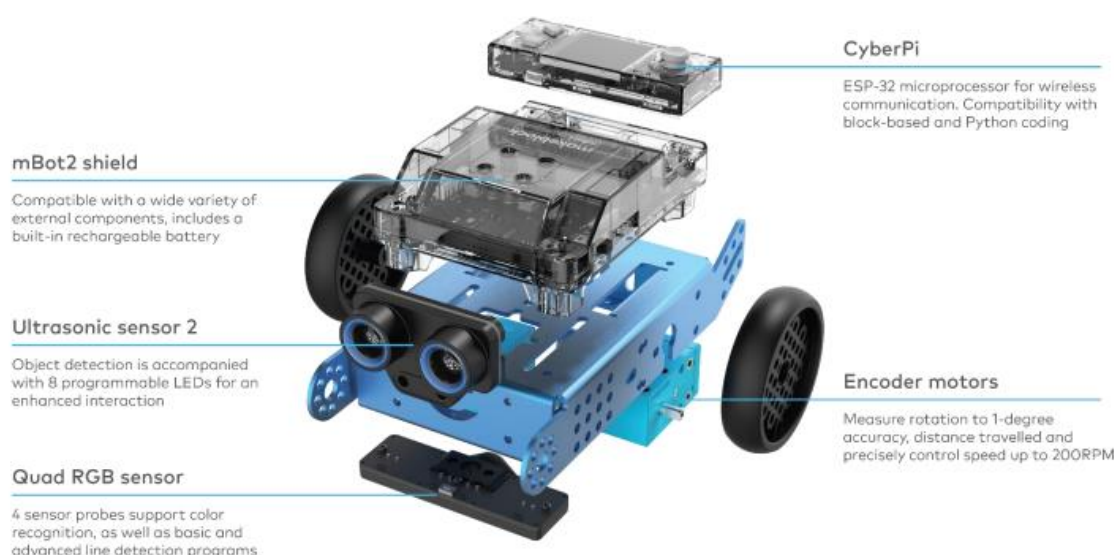


Fig. 25. Vista explosiva do mBot2, diferentes componentes presentes na sua constituição (Makeblock, EN mBot2 Getting Started Activities V1.1, 2022)

A locomoção do mBot2 é assegurada por motores com *encoders*. Com este tipo de equipamento, os alunos podem controlar de forma precisa a rotação, a velocidade, a posição das rodas e do robô. Adicionalmente, os motores podem ser usados como *servos*, e ainda como *knobs*, para devolver dados para o sistema, como se de um sensor se tratasse. A grande vantagem dos *encoder motors* é que permite uma maior integração de conceitos matemáticos no desenvolvimento de programas, como movimentar segundo distâncias precisas, calcular curvas e mapear um trajeto com a finalidade de percorrer um labirinto, transferindo os resultados para o computador.

Os sensores utilizados para efetuar o reconhecimento do ambiente ao seu redor estão realçados na Fig. 26: O sensor ultrassónico, a amarelo e o sensor seguidor de linha com capacidade de deteção de cor QuadRGB, a verde.



Fig. 26. Sensores para reconhecimento do ambiente, à esquerda o sensor ultrassónico, e à direita o sensor QuadRGB (adaptado de *Makeblock, 2022*)

O primeiro consegue realizar leituras mais precisas e consistentes do que as versões do mBot anteriores. Está localizado na parte frontal do mBot2, nos pequenos cilindros que se assemelham a “olhos”. Este sensor emite ultrassons e recebe o eco. Baseado no tempo que o som demora a ser devolvido para o recetor, o mBot2 calcula a distância do objeto.

O sensor ultrassónico está equipado com 8 LEDs azuis (Fig. 27 à esquerda). É possível programar estes LEDs para criar a ilusão de que o mBot é capaz de mostrar “emoções”, regulando a intensidade a que estes brilham. A intensidade pode ser controlada simultaneamente para todos os LEDs ou individualmente.



Fig. 27. Vista explodida do sensor ultrassónico (esquerda), realçando os 8 LEDs azuis na sua constituição e Sensor Quad RGB integrado no mBot2 (direita), realçando os sensores de luz e os LEDs RGB (Adaptado de Makeblock, EN mBot2 Getting Started Activities V1.1, 2022)

O sensor seguidor de linha, que também atua como sensor de deteção de cor, é equipado com quatro sensores de luz e quatro LEDs RGB (conforme ilustrado na Fig. 27, à direita). Os sensores RGB são definidos como L1, L2, R1 e R2 (L para o lado esquerdo, R para o lado direito). Cada LED RGB emite uma luz que alterna rapidamente entre diferentes cores. Ao medir a intensidade da luz refletida para cada cor emitida pelo LED RGB, os sensores conseguem identificar a cor da superfície em que o mBot está localizado.

O processo de programação torna-se mais simples, porque o sensor consegue transmitir qual a cor detetada e não é necessário o utilizador verificar o código de cores RGB. O sensor consegue detetar seis cores diferentes, às quais se adicionam o preto e o branco. Contém ainda um botão na parte de cima do sensor que é utilizado para a calibração do mesmo (Makeblock, EN mBot2 Getting Started Activities V1.1, 2022).

O kit mBot2 inclui uma pista de teste (apresentada na Fig. 28) que é desenhada para testar as funcionalidades do sensor seguidor de linha. Esta pista possibilita a realização de várias atividades que exploram as funções do sensor.

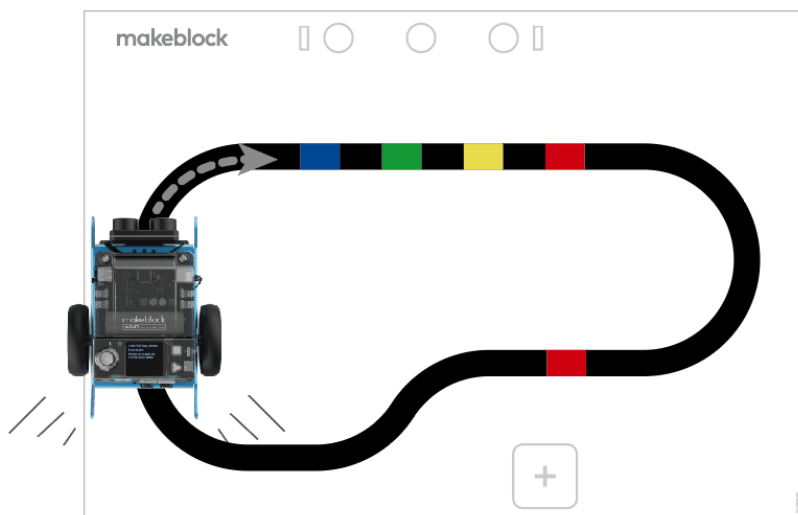


Fig. 28. Esquema da pista de teste incluída no kit mBot2 (Makeblock, 2022).

3.1.2. Ultimate 2.0

O Ultimate 2.0 (Fig. 29) é uma versão mais recente do Ultimate 1.0 apresentado anteriormente, também criado pela plataforma Makeblock. Contém mais de 550 peças mecânicas e módulos eletrônicos, permitindo ao utilizador, a construção de diferentes configurações. A placa de controlo MegaPi, responsável pelo tratamento e processamento de toda a informação, é baseada no chip ATmega2560. O MegaPi (Fig. 30), à semelhança do CyberPi do mBot2, é manipulado através do *software* mBlock. Consegue acionar *encoder motors*, motores DC e motores de passo. Tem a capacidade de acionar até 4 *motor drivers* (4 motores de passo ou 8 motores DC) e 10 engrenagens de direção ao mesmo tempo. Vários sensores podem ser conectados externamente. A capacidade máxima de saída de corrente e tensão é, respetivamente, 10 A e 12 V DC. O MegaPi suporta o Arduino IDE e programação gráfica (Scratch). Pode ser também combinado com um Raspberry Pi, permitindo a utilização de Python para controlar vários motores e módulos eletrônicos, expandido a possibilidade da construção de estruturas complexas como um carro robô, CNC ou até mesmo impressoras 3D (Fig. 29 à direita). Todas as extensões podem ser conectadas através de uma placa adaptadora MegaPi RJ25 (Fig. 30 à direita) (Makeblock, mBlock Block-Based Editor- Program a Device (Details)- Ultimate 2.0, 2022).



Fig. 29- Algumas configurações possíveis de construir com as peças disponíveis no kit Ultimate 2.0, desenvolvido pela Makeblock. Na direita, uma impressora 3D desenvolvida a partir das peças no kit, mais um conjunto de add-ons (sensores, display, etc) possíveis de implementar no MegaPi (Makeblock, Ultimate_senior, 2022).

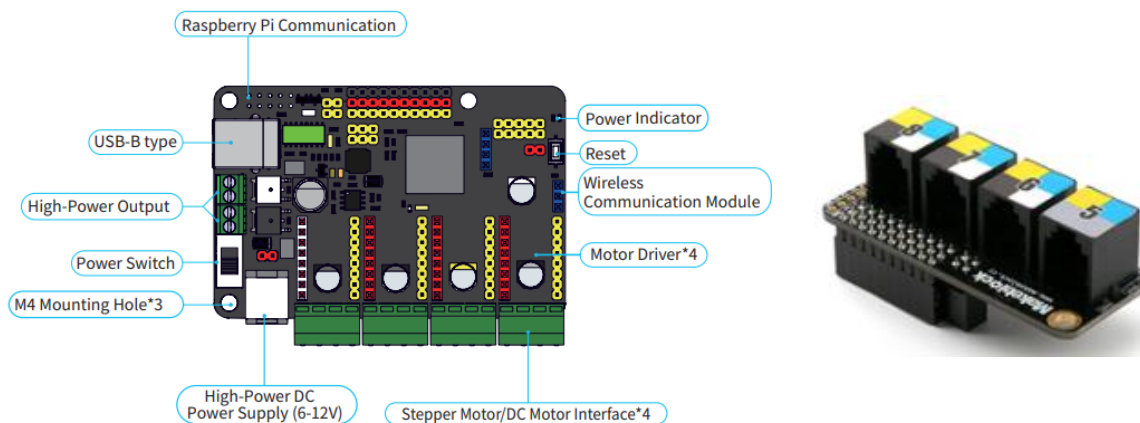


Fig. 30- Esquema de especificações técnicas da placa de controlo MegaPi, contida no kit Ultimate 2.0, desenvolvido pela Makeblock. Na direita, a placa adaptadora MegaPi RJ25, utilizada para converter os pinos do MegaPi para a interface RJ25 (Makeblock, Ultimate_senior, 2022)

A nível sensorial, o kit contempla um sensor de linha constituído por dois sensores.

Cada sensor tem um LED emissor de infravermelho e um foto transístor de deteção. Permite assim que o robô se possa mover ao longo de uma linha preta num fundo branco ou vice-versa. Está equipado com conetores S1, S2, VCC e GND, tornando possível a instalação na maioria das

placas de controlo (Arduíno) (Fig. 31 à esquerda), a outra possibilidade de conexão é através da porta RJ25 (Makeblock MegaPi e Makeblock Orion) (Fig. 31 à direita).

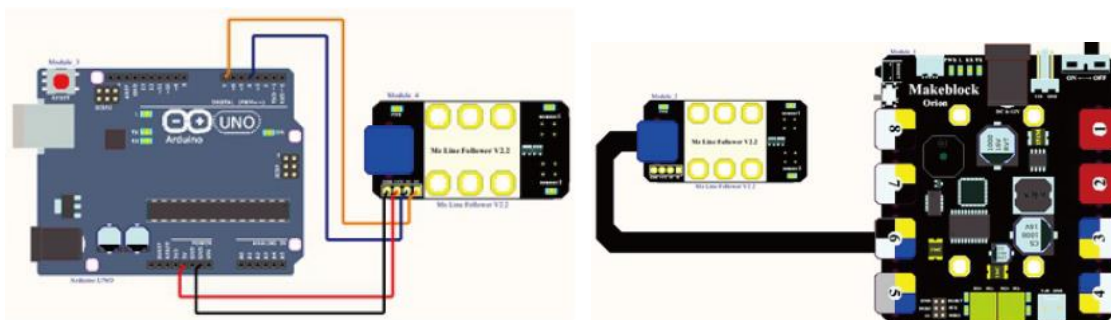


Fig. 31- Conexão do sensor de linha incluído no kit Ultimate 2.0 com um Arduíno Uno (à esquerda), conexão do sensor de linha com a placa de controlo Orion da Makeblock (à direita) (Makeblock, Ultimate_senior, 2022).

O sensor ultrassónico incluído no kit é idêntico ao do mBot2 (Fig. 26), capaz de medir distâncias entre os 3 e 400 cm. Assim como o sensor de linha, o sensor ultrassónico pode ser conectado através da interface RJ25 ou através dos conectores VCC, GND e SIG (sinal digital).

O giroscópio (Fig. 32 à esquerda) é um módulo de deteção de movimento e postura do robô. Contém um acelerómetro de 3 eixos, sensor de velocidade angular de 3 eixos, um processador de movimento e comunicação de interface I²C. Suporta programação Arduíno IDE e ainda programação gráfica mBlock 3 (Scratch). A conexão pode ser realizada através da interface RJ25 ou através dos pinos SCL, SDA, VCC, GND (para placas de controlo Arduíno por exemplo).

O módulo *shutter* (Fig. 32 no centro) é projetado para tirar fotografias de forma automática através de câmaras SLR. Pode ser utilizado para controlar a exposição de tempo da captura fotográfica ou do vídeo (*time-lapse*).

O MegaPi Encode/DC Mux Driver (Fig. 32 à direita) é um módulo capaz de acionar 2 motores DC ou 1 *encoder motor* (Fig. 33). Adota o modo plug-in de 2 x 8 pinos e pode ser facilmente instalado no MegaPi.



Fig. 32- Alguns componentes incluídos no kit Ultimate 2.0, giroscópio, módulo shutter e MegaPi Encode/DC Mux Driver, respetivamente, da Makeblock.



Fig. 33- Encoder motor 25 DC 6V/185 RPM, controlado pelo MegaPi Encode/DC Mux Driver (Makeblock, Ultimate_senior, 2022).

Para além dos componentes contidos no kit Ultimate 2.0, já foi mencionado a possibilidade de adição de extensões. A Fig. 34, ilustra alguns dos módulos desenvolvidos pela Makeblock (mBuild) que podem ser implementados em configurações diferentes do Ultimate 2.0, desde sensores de gás, sensores de luz, display, potenciômetros, etc. Ao nível da programação, a plataforma mBlock contém bibliotecas com os respetivos comandos de manipulação destas extensões.

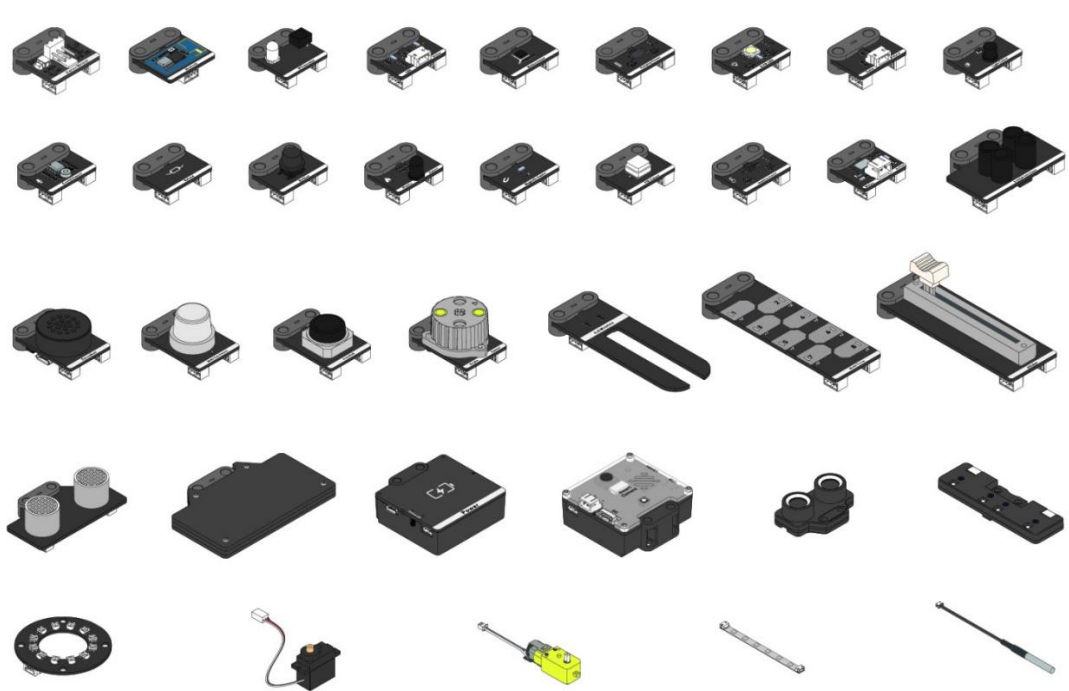


Fig. 34- Módulos mBuild e estruturas de suporte desenvolvidos, com o objetivo de expandir as possibilidades de programação dos robôs Makeblock (Makeblock, Ultimate_senior, 2022).

Um dos módulos explorado ao longo do trabalho desenvolvido nesta dissertação, foi a Smart Camera (Fig. 35). Este módulo desbloqueia a possibilidade de criar protocolos que envolvam Visão Computacional, *Machine Learning*, abrindo uma janela de oportunidades de implementação de Inteligência Artificial nos robôs Makeblock.



Fig. 35. Smart Camera, módulo disponibilizado pela plataforma Makeblock que possibilita a criação de programas que envolvam Visão Computacional e Inteligência Artificial (Makeblock, Smart Camera, 2022).

3.1.3. Software de programação mBlock 5

O mBlock 5 (exemplo da janela de programação na Fig. 36) foi projetado para a educação em Ciência, Tecnologia, Engenharia, Artes e Matemática (STEAM). Inspirado no Scratch 3.0, suporta linguagens de programação gráficas e textuais. Permite a criação de projetos próprios, jogos, animações e programar dispositivos como robôs Makeblock e Microbit. O mBlock 5 fornece dois editores, isto é, o editor baseado em blocos e o editor Python, além disso, integra tecnologias como Inteligência Artificial (IA) e Internet of Things (IoT) (Makeblock, mBlock Block-Based Editor-About mBlock5- What is mBlock 5 Block-Based Editor, 2022).

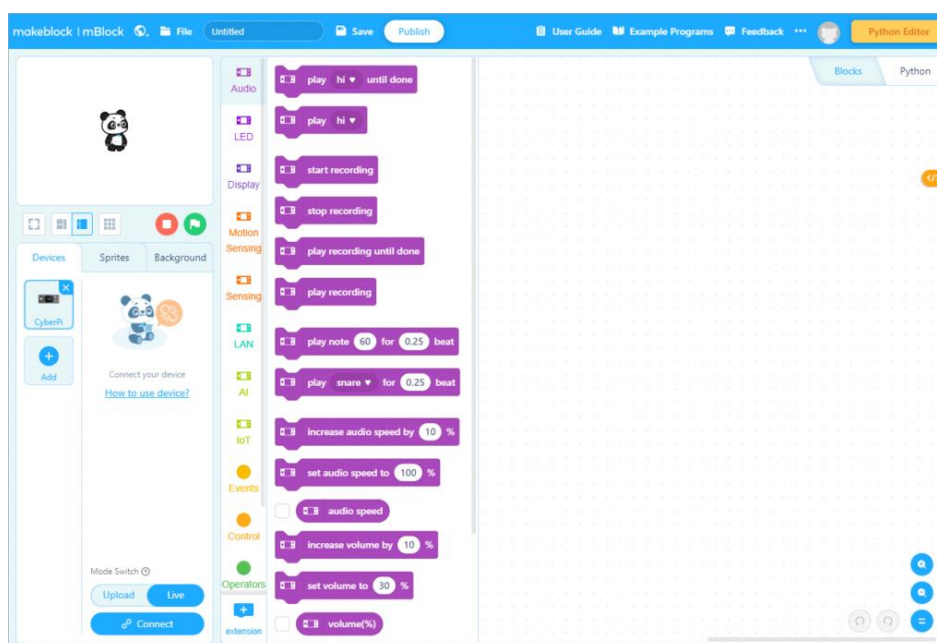


Fig. 36. Ambiente de programação Scratch, disponibilizado pelo software mBlock 5, desenvolvido pela plataforma Makeblock.

Scratch

O editor baseado em blocos (Scratch) (Fig. 37) é o editor predefinido no mBlock 5. Subdivide-se em 3 áreas de trabalho (Stage area, Blocks area, Scripts area). A Stage area é a área onde podemos selecionar os desenhos, conectar dispositivos, definir os sprites (objeto gráfico bidimensional que se move na tela em função do programa) e os respectivos cenários. Na Blocks area, encontram-se os blocos em função de categorias e cores. Por último, a Scripts area é a área onde é possível compilar o programa arrastando os blocos.

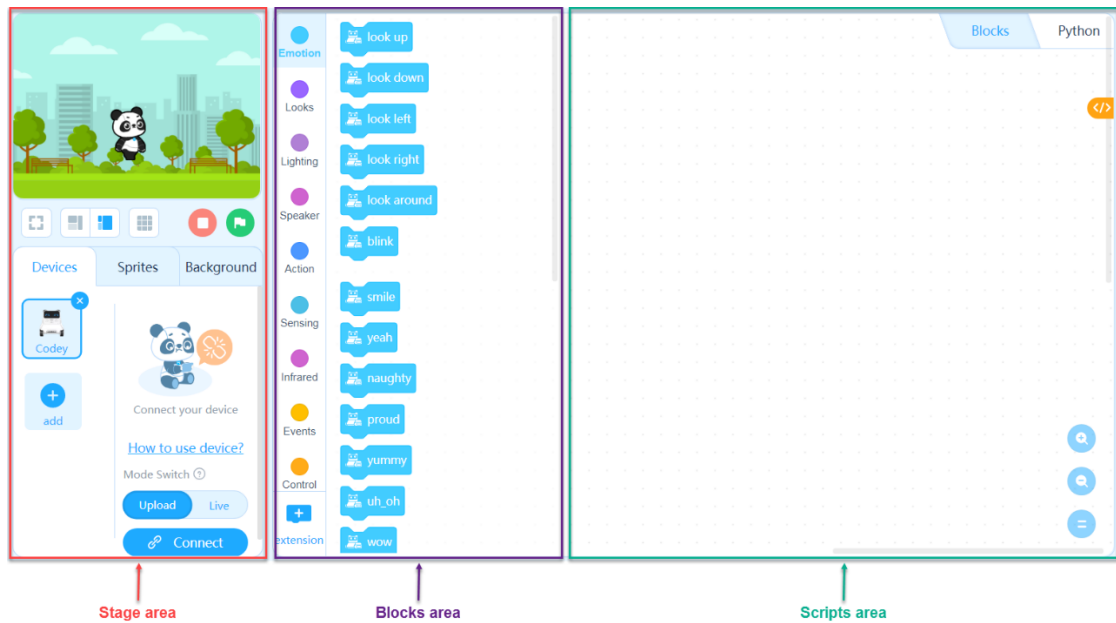


Fig. 37. Editor Scratch, disponível no software mBlock 5. Subdividido em 3 principais áreas de trabalho: Stage area; Blocks area e Scripts area (Makeblock, mBlock Block-Based Editor-About mBlock5- What is mBlock 5 Block-Based Editor, 2022).

Python

Em 2018, um editor de código (Fig. 38) foi incorporado ao mBlock 5 como um complemento à programação baseada em blocos, com o objetivo de ajudar os utilizadores a realizar a transição da programação baseada em blocos para a programação baseada em linhas de código. O editor suporta a transcodificação em tempo de real de blocos e códigos, permitindo assim a aprendizagem de Python.

Em 2020, a Makeblock passou de um editor capaz de transcodificar, para um editor Python, procurando assim, satisfazer as necessidades educacionais cada vez mais difundidas e profissionais.

Atualmente, o editor Python fornece um ambiente Python real, permitindo a utilização de várias bibliotecas de terceiros. Além disso, suporta a programação de *hardware*, trazendo assim uma maior diversidade de cenários de programação e adaptação para os principiantes, aplicável a vários cursos de Python como Inteligência Artificial (AI) e Análise de Dados.

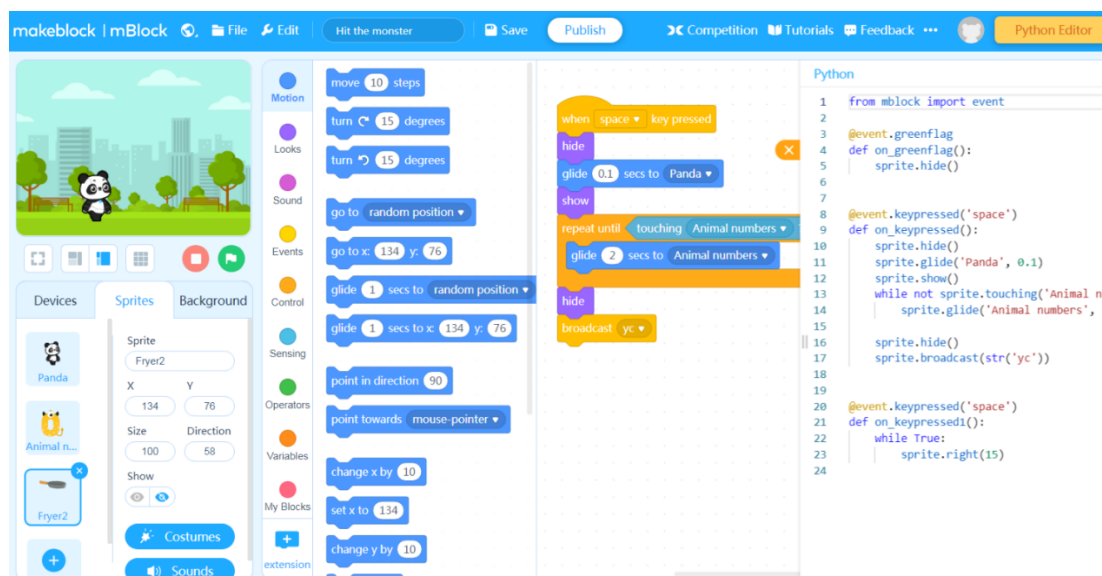


Fig. 38. Editor inicialmente desenvolvido pela Makeblock, capaz de transcodificar um programa em Scratch para Python (Makeblock, mBlock Python Editor- About mBlock 5 Python Editor- What is mBlock-Python Editor, 2022).

Contudo, a Makeblock procura continuamente melhorar o editor Python e interagem com a comunidade de programadores de robôs Makeblock para o conseguir (Makeblock, mBlock Python Editor- About mBlock 5 Python Editor- What is mBlock-Python Editor, 2022).

Arduíno

Para o kit Ultimate 2.0, a placa de controlo MegaPi foi desenvolvida com base num microcontrolador ATmega2560, compatível com programação Arduíno. As principais diferenças entre o MegaPi e um Arduíno (seja ele Mega, UNO, etc) centram-se nos conetores, o MegaPi usa conetores RJ45 em vez de pins. Como todos os módulos disponibilizados pela Makeblock utilizam conetores RJ45, torna a conexão dos módulos bastante fácil, sem necessidade de soldar fios individuais (Candelas *et al.*, 2016). Uma vez que o Arduíno IDE é amplamente usado pela comunidade educativa, existem muitos cursos disponíveis que permitem a rápida aprendizagem de C++, e o desenvolvimento de projetos na área da robótica (Sobota *et al.*, 2013).

O mBlock 5 contém também um editor Arduíno quando adicionada a extensão do dispositivo Ultimate 2.0, oferece ainda a possibilidade de transcodificação de Scratch para Arduíno (Fig. 39). Permitindo a aprendizagem através do processo de engenharia reversa.

Para além de programar o MegaPi, o mBlock5 permite programar diretamente uma placa Arduino Mega, Arduino Uno, RaspberryPi adicionando os dispositivos na Stage area (Fig. 37).

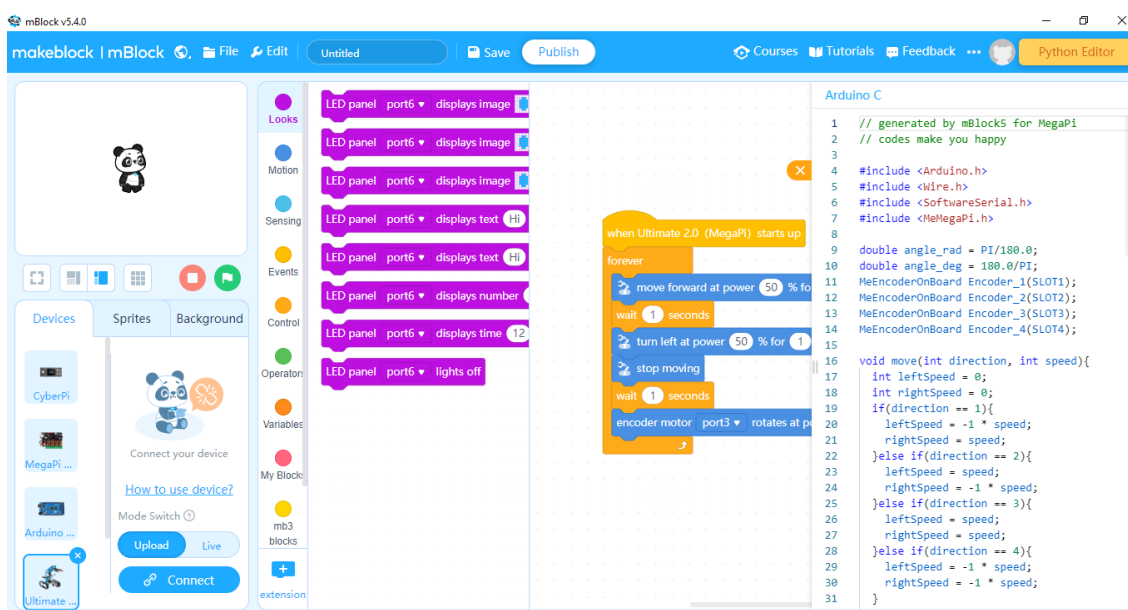


Fig. 39. Transcodificação de um código em Scratch para Arduino, no software mBlock 5. O código manipula os motores instalados no kit Ultimate 2.0, projetado pela plataforma Makeblock.

3.2. Processos de fabrico

Durante a preparação e realização das atividades descritas ao longo desta dissertação, existem alguns processos de fabrico inerentes, com principal destaque para o Fabrico Aditivo. A maioria dos trabalhos contém suportes, peças ou infraestruturas produzidas através de impressão 3D.

A mesa de trabalhos multiusos descrita no próximo capítulo contém também peças impressas, porém o corte das placas MDF foi feito através de Maquinação CNC no FabLab da Universidade da Beira-Interior.

Um dos trabalhos exequíveis na mesa é o labirinto, para a produção das paredes internas do mesmo foi cortada uma placa MDF recorrendo à tecnologia de corte a laser.

Nesta subsecção são abordados de forma breve as tecnologias e processos de fabrico que contribuíram para o sucesso daquilo que é o objetivo da dissertação, criar metodologias inovadoras para o ensino da robótica.

3.2.1. SOLIDWORKS

O SOLIDWORKS é um *software* de desenho eletromecânico que permite aos utilizadores desenvolverem ideias de forma criativa e rápida, produzindo modelos e desenhos detalhados. É um software abrangente e eficiente, que auxilia no projeto e fabricação de produtos (Onwubolu, 2017).

A capacidade de projetar, simular testes elétricos e comportamentos mecânicos caracterizam a multidisciplinariedade deste *software*, proporcionando um fluxo de trabalho contínuo e integrado durante todas as fases de desenvolvimento de um produto: projeto, verificação, design, comunicação e gestão de dados (Veiros, 2020).

3.2.2. Maquinação por Comando Numérico Computorizado (CNC)

O Controlo Numérico Computadorizado (CNC) é um método de controlo de máquinas que utiliza um software para pré-programar a operação de ferramentas em equipamentos com múltiplos eixos. Este processo é aplicado para controlar uma ampla variedade de máquinas, incluindo centros de maquinação, tornos, máquinas de corte a jato de água, fresadoras, corte a laser e outros.

A linguagem de programação utilizada para manipular o controlador da máquina CNC é código G. A partir da leitura do script, o controlador controla o movimento da máquina. O código pode ser criado pelo operador e inserido manualmente, ou gerado a partir de um *software* CAM (*Computer Aided Manufacturing*) (Frank *et al.*, 2004).

Para a construção das infraestruturas de apoio à robótica movel desenvolvidas no âmbito deste trabalho, foram utilizados alguns métodos de manufatura subtrativa por CNC utilizando alguns equipamentos do Laboratório de Fabricação da Universidade da Beira Interior.

Router CNC de 3 eixos

A Router CNC de 3 eixos é um equipamento amplamente utilizado para produzir peças em diversas indústrias como marcenaria, fabricação de protótipos, entre outras. A Router CNC utilizada no presente trabalho, possui três eixos: X, Y, Z.

O eixo X controla o movimento horizontal da máquina, permitindo que a ferramenta se desloque ao longo do comprimento da peça de trabalho. O eixo Y controla o movimento vertical da máquina, movendo a ferramenta ao longo da largura da peça. Por fim, o eixo Z controla a profundidade do corte ou gravação (Frank *et al.*, 2004).

Corte a laser

O processo de corte a laser CNC implica a utilização de um feixe de laser controlado por computador para efetuar cortes em diversos materiais, tais como metal, madeira, plástico, entre outros. O computador regula com precisão o movimento do feixe de laser de acordo com um caminho pré-estabelecido, criado a partir de um desenho CAD. Esta tecnologia possibilita a realização de cortes detalhados e complexos, sendo extensivamente utilizada em múltiplas indústrias, abrangendo setores como fabricação, indústria automóvel, eletrónica, entre outras.

3.2.3. Fabrico Aditivo (Impressão 3D)

A impressão 3D é um processo de fabrico aditivo que utiliza código G, obtido a partir do processamento de um modelo *software* de CAD para criar um objeto através da deposição de material, camada após camada, executando as formas geométricas pretendidas. Existem várias tecnologias de fabrico aditivo, sendo que a principal diferença entre os diversos processos reside na forma como as camadas são geradas para criar peças (Veiros, 2020).

O processo de impressão 3D utilizado para a produção de peças e suportes nesta dissertação foi a FFF - *Fused Filament Fabrication*, em português, fabricação por filamento fundido). O processo é baseado na extrusão de filamentos termoplásticos, sendo estes aquecidos através de uma extrusora, e através de um bucal, depositados em camadas numa plataforma (cama). A simplicidade, fiabilidade e acessibilidade do processo FFF tornaram a tecnologia de fabricação aditiva amplamente reconhecida e adotada na indústria, pelo meio académico, investigadores e pelos consumidores (Ngo *et al.*, 2018).

Existem vários tipos de materiais utilizados na impressão 3D FFF. Os mais comuns são os plásticos, como ABS, PLA, PETG, os dois últimos bastante utilizados no envelope de trabalhos desenvolvidos nesta dissertação devido às suas propriedades mecânicas e custo reduzido.

4. Infraestrutura de Apoio à Robótica Móvel

4.1. Mesa de trabalho multiusos

Todos os trabalhos descritos ao longo desta dissertação exigiram a criação de estruturas ou cenários para que os robôs pudessem executar os desafios propostos, desde a resolução de um labirinto, até a um jogo de futebol.

Uma das preocupações iniciais sempre foi ter um local no laboratório de Robótica destinado à robótica móvel. Esse local teria de ser seguro para os robôs se movimentarem sem existir risco de colisões ou quedas que pudessem resultar em consequências irreversíveis.

O espaço definido para a robótica móvel foi uma mesa com uma área de trabalho de 2300 x 1200 mm. Para fazer um protótipo desta bancada de trabalho, planeou-se a fixação de placas de madeira em torno da bancada, funcionando como proteções laterais, para que os robôs não saíssem da área estabelecida.

As placas de madeira não podiam ser coladas ou aparafusadas, pois isso ia resultar em alterações permanentes na mesa, logo a solução idealizada para fixar as proteções laterais foi o desenvolvimento de grampos recorrendo à impressão 3D.

Várias versões de suportes foram projetadas no Solidworks. Na Fig. 40 é possível observar algumas das versões destes suportes. Inicialmente pensou-se numa peça que contornava a parte exterior do tampo da mesa e fixava na parte inferior com auxílio de um parafuso. A rosca era criada com a utilização de um macho. Porém, a força aplicada do parafuso criava uma flecha ao longo da peça.

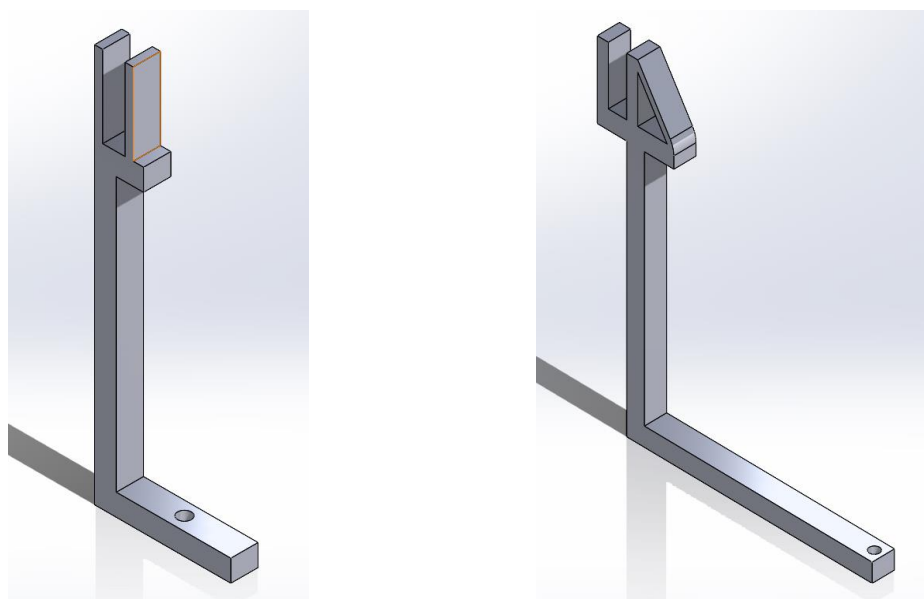


Fig. 40. Versão 1 e 2, respetivamente, do suporte das proteções laterais da mesa dedicada à robótica móvel.

A segunda versão ainda tinha a aplicação do parafuso na extremidade da peça. O problema da flecha mantinha-se e a peça não apresentava solidez suficiente quando era exercida força sobre a placa de madeira que se encontra fixa nas ranhuras do suporte.

Seguiu-se a tentativa de aproveitar melhor a geometria do tampo da mesa e desenvolveu-se uma peça que encaixava na parte inferior do tampo sendo posteriormente colada à peça que suportava a placa (Fig. 41). Novamente, revelou ser uma solução que não se adequava, pois o suporte cedia na zona de união das peças.

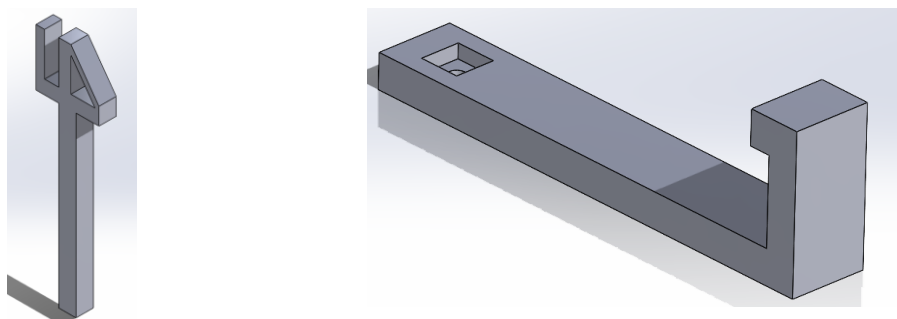


Fig. 41. Versão 3 do suporte das proteções laterais da mesa dedicada à robótica móvel

Por último, a versão que melhor funcionou continha 3 peças, uma estrutura cuja extremidade era uma superfície roscada, uma peça que apoiava na parte inferior do tampo da mesa, e uma porca que fixava o conjunto todo. Na Fig. 42 é visível o desenho CAD do suporte, bem como a sua aplicação na mesa (Fig. 43).

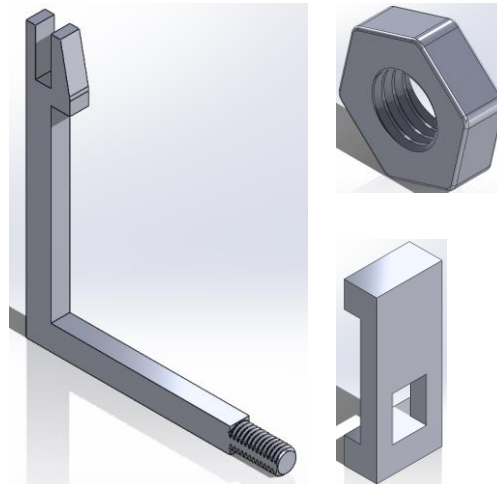


Fig. 42. Versão final do suporte das proteções laterais da mesa dedicada à robótica móvel.



Fig. 43. Versão final do suporte já com a proteção lateral aplicada na mesa.

Após a realização de atividades com alunos nesta bancada, e a observação do seu funcionamento, partiu-se do protótipo para o desenvolvimento de uma solução mais robusta e multifacetada capaz de englobar todos os trabalhos como labirintos, simulações de situações de trânsito, Futebot e

outro tipo de ideias que possam surgir em trabalhos futuros. Uma estrutura robusta, com a possibilidade de adicionar ou retirar componentes em função da atividade que nela é realizada.

Iniciou-se um novo projeto, uma nova versão da mesa de trabalho dedicada aos projetos e atividades que envolvem robótica móvel, juntando a possível integração de projetos da unidade curricular de Automação Industrial como por exemplo o projeto dos semáforos.

Para que fosse possível implementar as paredes do labirinto, a base da mesa teria de ter algum mecanismo que permitisse a adicionar/remover suportes. A solução desenvolvida foi um mecanismo de encaixe com molas. Para produzir o suporte, recorreu-se à impressão 3D. O mecanismo de encaixe é constituído por 2 peças como é possível observar na Fig. 44.

Contudo, a peça onde a coluna encaixa tem de permanecer fixa na mesa. Para alcançar o fim desejado, foi desenhada um suporte que estará embutido na placa MDF (16 mm de espessura) que servirá de base. A peça (Fig. 45) tem secções diferentes para não ser possível removê-la quando o suporte é puxado. Posteriormente foi feito o desenho planificado das placas que constituem a bancada, e foi feita a montagem para se verificar a compatibilidade entre peças. Os recortes nas extremidades facilitam a fixação das placas laterais. Na Fig. 46 é possível observar o desenho CAD da base, com os rasgos que apresentam uma redução de secção para fixar a peça presente na Fig. 45.

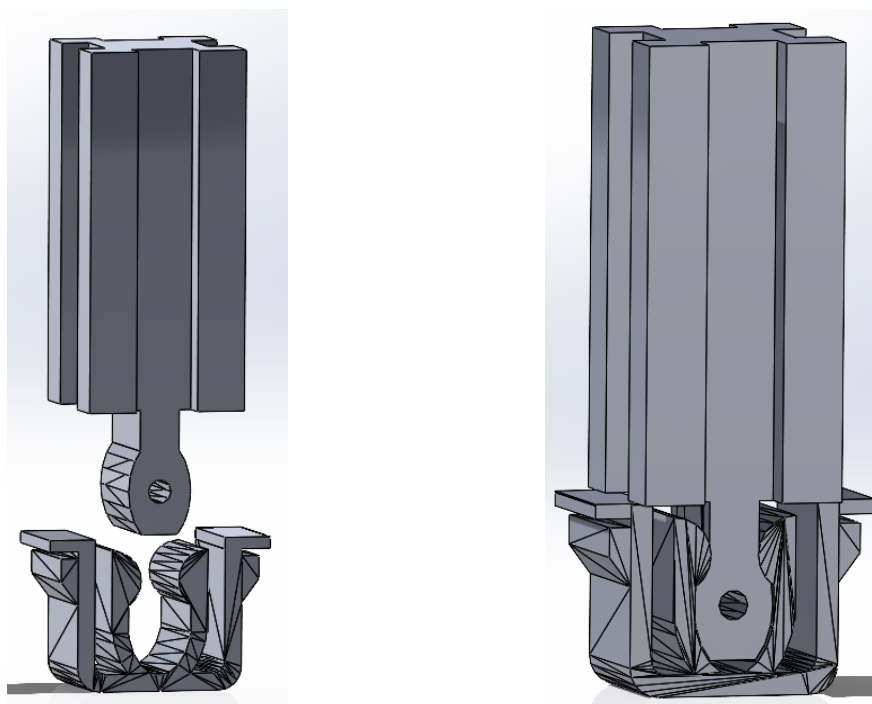


Fig. 44. Mecanismo de encaixe à pressão desenvolvido em Solidworks.

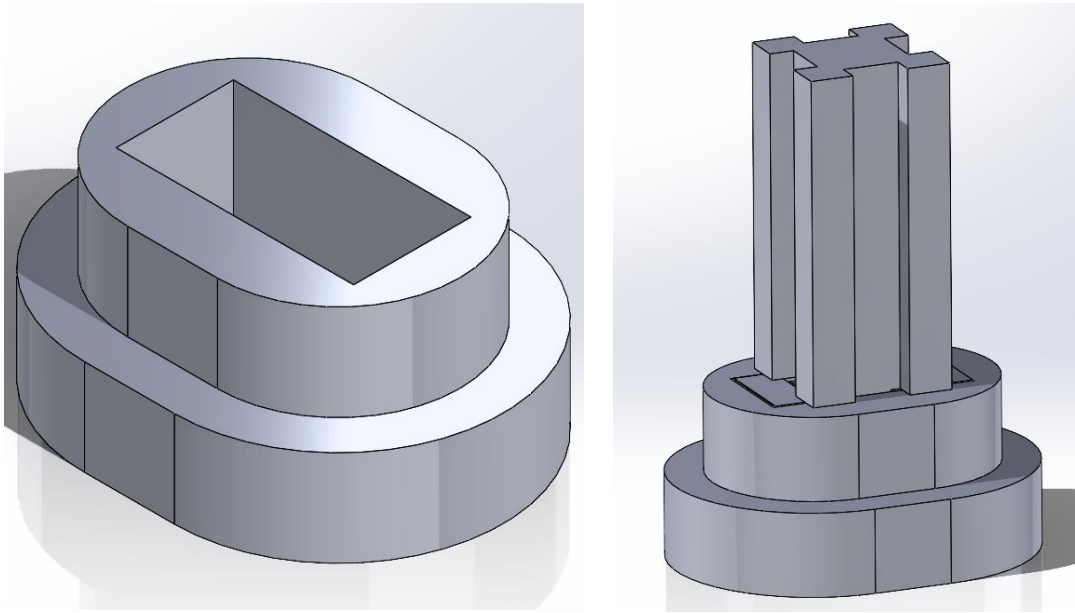


Fig. 45. Desenho CAD da peça embutida na base da mesa de trabalho.

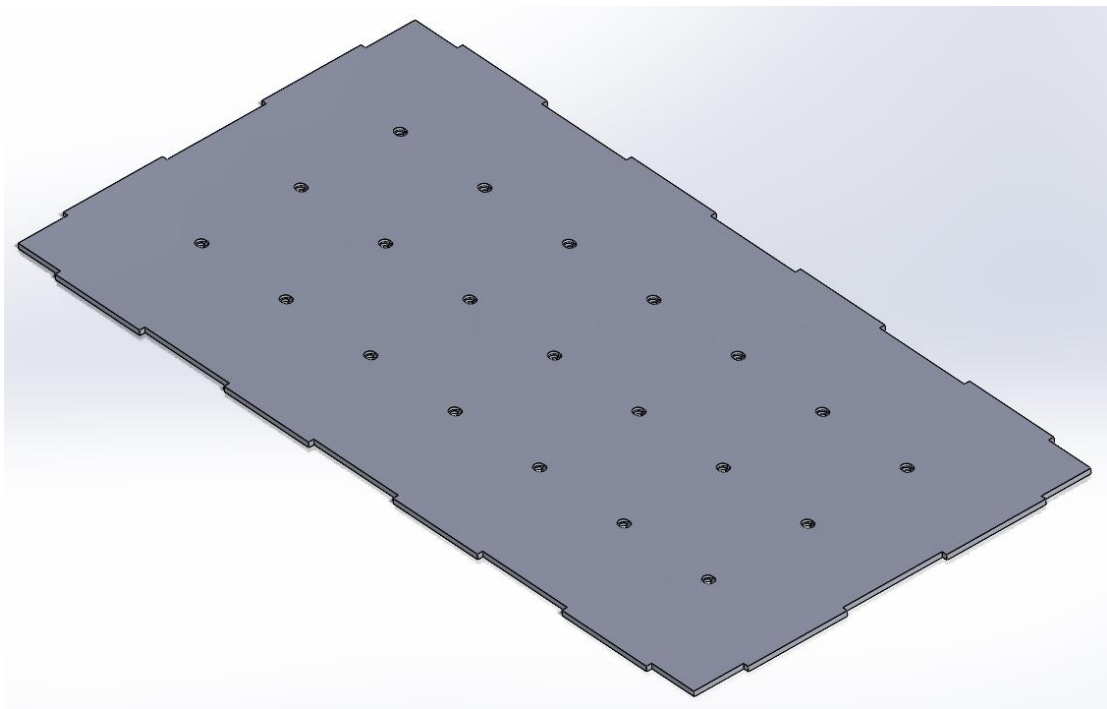


Fig. 46. Desenho CAD da base da mesa de trabalho.

As placas laterais (Fig. 47) apresentam recortes compatíveis com os da base. É possível observar também rasgos com espaçamento de 300 mm entre si. Nesses rasgos serão colocadas peças onde serão apoiadas as paredes internas do labirinto.

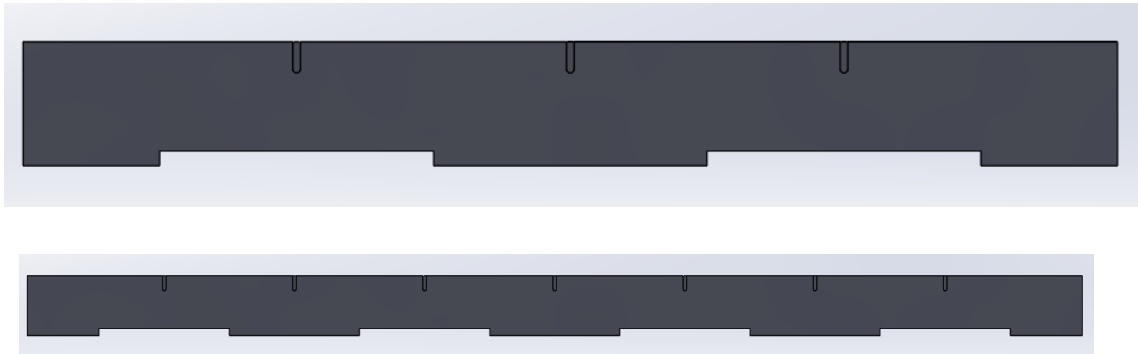


Fig. 47. Desenho CAD das placas laterais da mesa de trabalho.

O corte das placas MDF foi feito através de Maquinação CNC no FabLab da Universidade da Beira-Interior. As placas laterais foram coladas e aparafusadas na base.

O acabamento final da mesa foi cuidadosamente realizado. Após minuciosos preparativos foi dado acabamento à superfície de corte e todas as peças foram coladas, e aparafusadas. Posteriormente, foi feita a lixagem e nivelamento da superfície. Quanto ao revestimento, aplicou-se um tapa-poros. Essa etapa crucial permitiu preencher pequenas imperfeições e criar uma base uniforme para a aplicação da pintura posterior. A Fig. 48 ilustra as diferentes etapas de preparação da mesa.



Fig. 48. Fase de acabamento da mesa de trabalho, à esquerda ainda no processo de passagem do tapa-poros sobre a superfície. À direita, a mesa após o processo de pintura e já com algumas placas do labirinto colocadas nos suportes.

Para as paredes do labirinto utilizou-se placas de MDF com espessura de 5 mm, cortadas com precisão utilizando tecnologia de corte a laser Fig. 49. Estas garantem resultados extremamente precisos, permitindo que as placas encaixassem perfeitamente, criando um labirinto robusto e bem estruturado. Além disso, a utilização do MDF como material proporciona resistência e durabilidade.

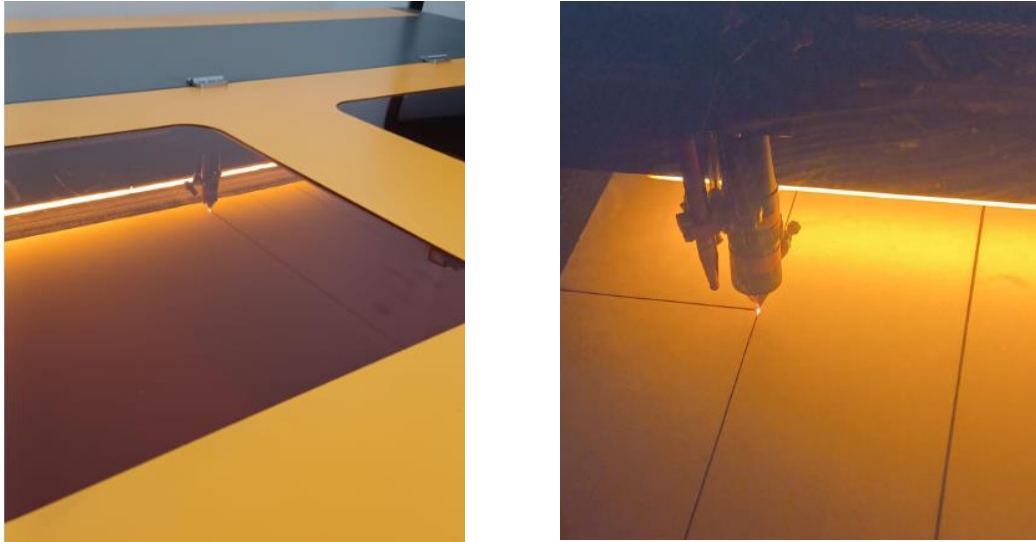


Fig. 49. Corte a laser da placa MDF para a produção das paredes do labirinto.

4.2. Futebot

O Futebot é um torneio de futebol realizado com os mBot2. Embora seja uma atividade que não exige qualquer tipo programação, foi necessário criar uma estrutura que possibilite a realização de jogos em segurança, e evitando danos nos robôs.

Aproveitando a mesa de trabalho multiusos, criaram-se balizas com um design compatível às proteções laterais (Fig. 50 à esquerda), recorrendo à impressão 3D.

Foram projetados para-choques flexíveis para proteger os mBots, desenhados com o objetivo de absorver a energia gerada por colisões entre eles ou contra as paredes laterais da mesa (Fig. 50 à direita). As peças foram produzidas recorrendo novamente à impressão 3D e aparafusadas nos furos existentes no chassi do robô. O para-choques frontal tem a função acrescida de possuir um encaixe que facilita a manipulação da bola.

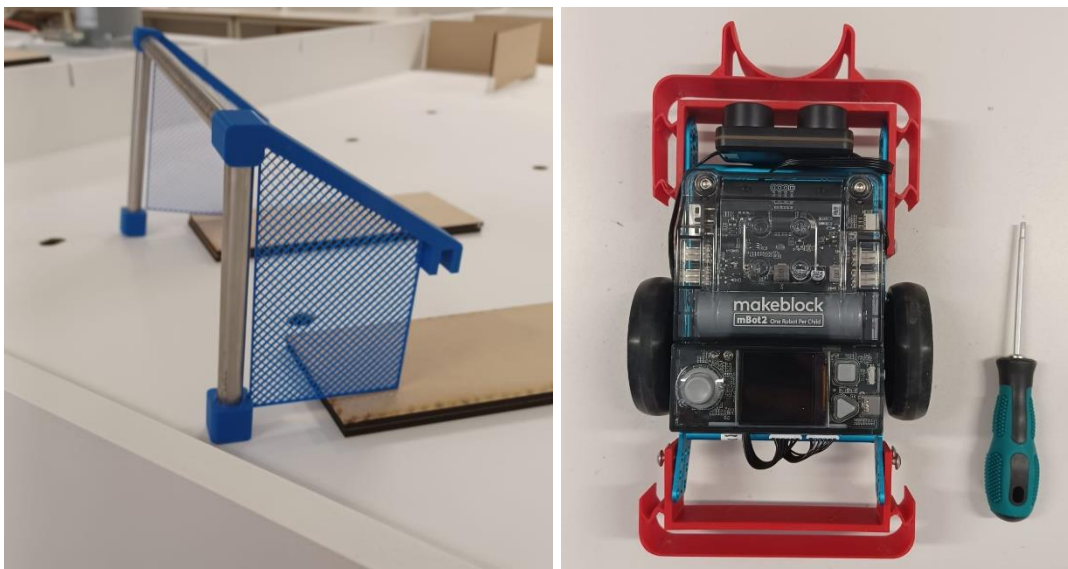


Fig. 50. Balizas desenvolvidas para a atividade do Futebot (esquerda) e montagem dos parachoques no mBot2 (direita).

4.3. Garra para mBot2

Durante as primeiras atividades realizadas com o mBot2, foi reconhecido o enorme potencial da criação de *add-ons* que poderiam dar uso às portas de input/output do mBot2 e ser implementados para a realização de protocolos e trabalhos futuros.

O primeiro componente desenvolvido neste sentido, foi uma garra controlada por um servomotor, que conferiu ao mBot2 capacidades de manipulação, que não estavam incluídas no kit original. O desenho da peça foi realizado em SOLIDWORKS (Fig.51 à esquerda), e a sua produção recorrendo à impressão 3D. Na Fig. 51 (à direita), é possível observar o robô com a garra instalada durante a realização de uma atividade cujo objetivo era mover um objeto.

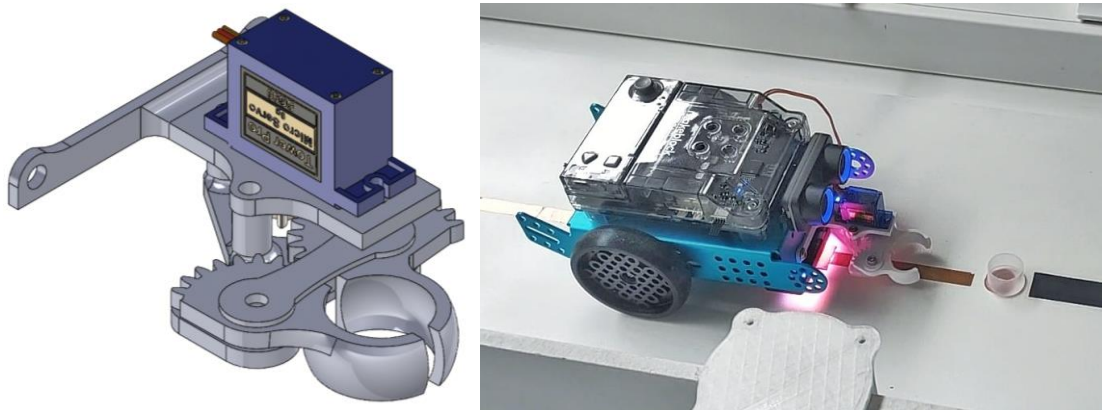


Fig. 51. CAD da garra desenvolvida (esquerda).e mBot2 no processo de manipulação de objetos (direita)

Esta garra, também faz uso da furação roscada existente no chassi do mBot2 para permitir o fácil e rápido acoplamento do *add-on*.

5. Atividades e Recursos Educativos de Robótica

5.1. Formação com mBot2

Neste subcapítulo apresenta-se o processo de planificação de uma intervenção realizada no âmbito das atividades do projeto UBImpulso, *Green and sustainable growth in a digital world*, dirigida ao ensino da robótica por via de kits de robótica móvel com os alunos de uma escola profissional.

Nesta formação, que teve a duração de 3 aulas de 2 horas para cada uma das turmas, optou-se pela aplicação de uma metodologia de aprendizagem baseada na resolução de problemas (ABRP), em que os alunos tinham de resolver vários problemas, sendo aumentado o grau de dificuldade, sempre com a dinâmica de trabalho de grupo, promovendo o espírito crítico e o trabalho colaborativo e a socialização entre eles.

Durante as aulas lecionadas, o aluno teria de ser o agente do seu próprio desenvolvimento intelectual, desenvolvendo a análise crítica, o raciocínio e a criatividade. Sendo apenas dado orientação quando necessária.

Esse apoio aos alunos foi feito através de questões sugestivas e indicações úteis na clarificação dos desafios, ou recolocação do objetivo de forma a atingir o resultado esperado.

Os alunos tinham de procurar uma solução tendo em conta experiências anteriores, serem capazes de detetar erros, isolá-los e corrigi-los. Procurou-se elaborar problemas desafiantes, capazes de proporcionar reflexão e que cativassem a cooperação.

Na planificação da formação, o total de aulas foi dividido em 3 aulas de 2 horas para cada uma das turmas, iniciando com uma aula na qual se introduzia o kit robótico e explicava o funcionamento de aspetos básicos da programação do mesmo. Após o primeiro contacto, seguiam-se desafios por ordem crescente de complexidade de forma a preparar os alunos para o desafio final.

5.1.1. Aula 1- Introdução ao mBot2

Na primeira aula houve uma breve apresentação da Universidade aos participantes. A seguir, houve uma introdução à robótica, como podem os robôs auxiliar os seres humanos a desempenhar tarefas e quais os contextos onde se inserem.

Os alunos preencheram o primeiro questionário, um conjunto de questões que permitiam perceber se já teria existido algum contato com robótica, programação, sensores e outro conjunto de elementos constituintes de um robô.

De seguida introduziu-se o mBot2, quais os componentes mecânicos que o kit contém, o tipo de sensores e o seu princípio de funcionamento de forma sucinta. Falou-se ainda da panóplia de extensões possíveis de adquirir e implementar no mBot2.

Enquanto os participantes instalavam a aplicação mBlock nos computadores, foi dada uma noção do que é um algoritmo, que tipo de comandos existem no Scratch e quais as suas funcionalidades. Durante esta fase, procurou-se aprofundar essencialmente os comandos necessários para a realização do conjunto de atividades planeadas para a primeira aula.

O objetivo da primeira atividade passava por fornecer um conjunto de instruções ao mBot2 que lhes permitissem percorrer um trajeto sem utilizar qualquer tipo de sensor. Os alunos tinham de tentar perceber qual o conjunto de blocos que lhes permitia chegar ao objetivo através da perceção do espaço, arbitrando distâncias e o valor da rotação nas zonas da pista que continham curvas.

Sempre que os alunos apresentavam dificuldades na compreensão da funcionalidade de um determinado bloco, era realizada uma revisão do conteúdo do comando. Seguida da utilização dessa mesma instrução no robô para que o aluno pudesse associar um movimento físico à explicação prévia. Este processo foi similar em todas as aulas.

Após os grupos alcançarem o objetivo, foi pedido que pensassem em possíveis melhorias e realçou-se o papel dos sensores na robótica.

A atividade foi realizada novamente, mas desta vez, com auxílio de um sensor seguidor de linha instalado no robô. Esta alteração exigiu que os estudantes reformulassem todo o programa, pois todas as decisões que o robô teria de tomar, são em função dos valores transmitidos pelo sensor.

5.1.2. Aula 2- Conjugação de sensores

Na segunda aula foi realizada uma primeira introdução do funcionamento geral de um robô autónomo, que sensores são imperativos para o sucesso do desempenho de tarefas. Dando ênfase ao sensor de ultrassons no caso do mBot2.

Foram apresentados alguns exemplos de seres vivos que utilizam ultrassons, bem como exemplos na área da saúde.

De seguida, reviu-se o conteúdo lecionado na aula anterior e os estudantes tiveram a oportunidade de otimizar o programa com o objetivo de tornar o robô mais rápido na realização do trajeto, criando assim alguma competitividade entre grupos enquanto consolidavam conceitos.

Realizada a atividade introdutória da aula, foram introduzidos os blocos de programação que a extensão do sensor ultrassom disponibilizava e explicou-se aos alunos como criar um programa simples de parar a locomoção do robô quando detetasse um objeto. Aqui os alunos tiveram algum tempo para explorar as funcionalidades por eles próprios, testar, errar e tentar perceber a causa do erro.

Por fim, propôs-se aos alunos que colocassem o robô a seguir o trajeto da primeira aula, mas desta vez a pista teria obstáculos que seriam detetados pelo robô e teriam de ser contornados. A forma como os obstáculos eram contornados era definida pelo grupo, apelando à criatividade dos mesmos.

5.1.3. Aula 3- Labirinto e criação de variáveis

Na terceira aula realizou-se uma síntese dos conteúdos lecionados na aula anterior, quais os blocos de programação que melhor funcionaram para os diferentes grupos.

Posteriormente, os alunos retomaram a atividade anterior cujo propósito era contornar os objetos situados na pista. Desta forma, houve oportunidade de implementar melhorias e pensar em novas formas de navegação do robô.

De seguida, apresentou-se o cenário do labirinto (Fig. 52 esquerda), visto como o projeto final da formação. Os alunos tinham de construir um programa que completasse o labirinto desde uma posição de partida, até ser encontrada uma tira de fita vermelha no chão que simbolizava o final (Fig. 52 direita). Toda a estrutura do labirinto desenvolvida no âmbito dos trabalhos de Robótica Industrial foi reutilizada para a realização desta atividade.

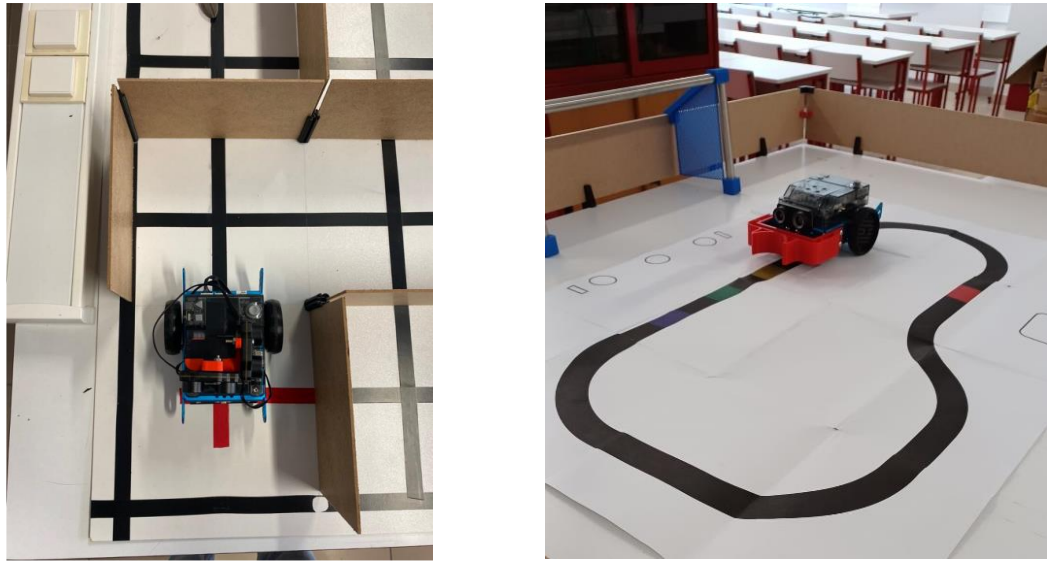


Fig. 52. mBot2 no fim do percurso do labirinto, destacado com fita vermelha na base (esquerda). mBot2 a percorrer a pista disponível no kit mBot2 (direita).

Visto que apenas está instalado um sensor ultrassom na parte frontal (ao contrário do que aconteceu no trabalho desenvolvido pelos alunos de Robótica Industrial que utilizaram dois sensores ultrassônicos), não é possível saber se existe alguma parede nas laterais do robô. Esta foi a principal barreira para os alunos que inicialmente procuravam virar o robô sempre na mesma direção, entrando em *loops* infinitos dentro do labirinto.

Eventualmente foi sugerido por um dos grupos que se criasse uma variável que contasse o número de curvas, e assim poderiam alterar a direção de rotação em função do número de curvas que o robô tinha realizado previamente (Fig. 53).

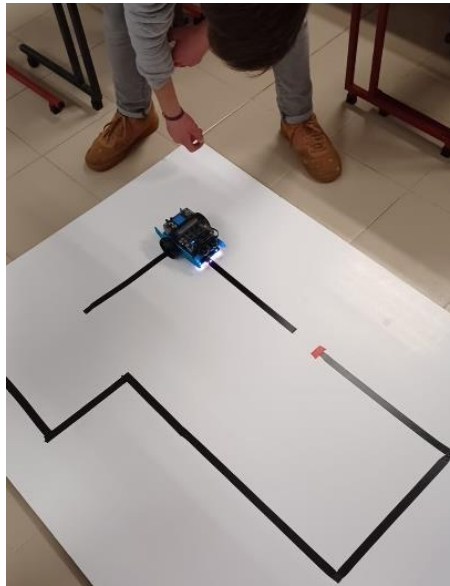


Fig. 53. Fotografia capturada durante o decorrer das atividades com os alunos da escola profissional.

5.2. Universidade de Inverno STEAM

No âmbito da 4ª edição da iniciativa Universidade de Inverno STEAM realizado na Universidade da Beira-Interior, 40 estudantes do 3º ciclo do Ensino Básico de vários pontos do país tiveram a oportunidade de visitar as instalações das diversas faculdades que constituem a universidade. As atividades decorreram durante quatro dias, iniciando-se na Faculdade de Engenharia. Os alunos foram divididos em 2 grupos de 20 elementos. Metade realizou a primeira atividade com os kits de robótica Lego Mindstorm, na qual assistiam uma luta de sumo entre dois robôs (Fig. 54).

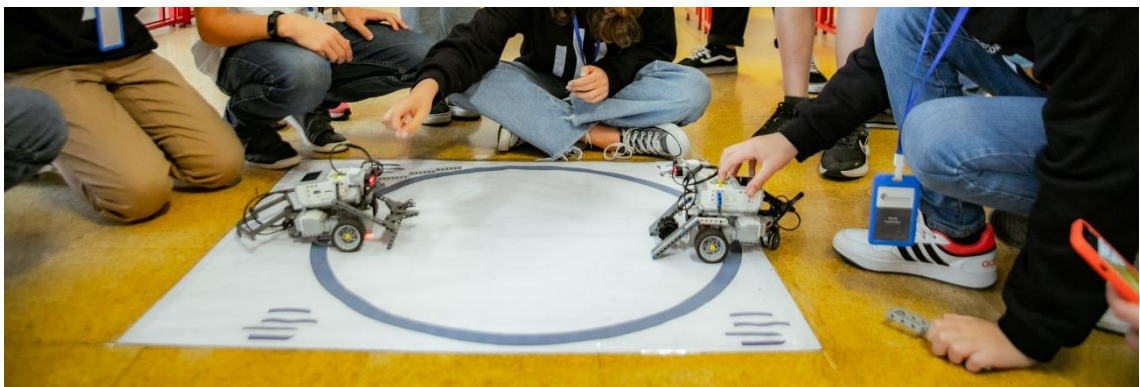


Fig. 54. Luta de sumo entre dois robôs Lego Mindstorm, atividade realizada no âmbito da iniciativa Universidade de Inverno STEAM que decorreu na Universidade de Beira Interior.

Os restantes alunos participaram num minitorneio de futebol de robôs, denominado Futebot. Esta atividade faz parte do envelope de trabalhos com kits de robótica móvel desenvolvido nesta dissertação.

Os alunos instalaram a aplicação *Makeblock* nos telemóveis, conectando-os aos *mBots* através de *Bluetooth*. A *Makeblock* disponibiliza a opção disponibilizadas de dirigir o robô de forma livre com o comando apresentado na Fig. 55.

Cada jogo tinha 4 participantes em simultâneo, distribuídos por 2 equipas (2x2), com um aluno por cada *mBot*. As partidas terminavam quando uma das equipas marcasse 2 golos na baliza adversária ou a duração da partida atingisse os 5 minutos.

A realização do Futebot exigiu a criação de um cenário, que incluísse duas balizas e uma linha de meio-campo onde a bola iria estar situada no início de cada partida.



Fig. 55. Display do modo “Dirigir” da aplicação *Makeblock*, à esquerda encontra-se o analógico que permite manipular a locomoção robô, os botões da direita realizam movimentos ou emitem sons predefinidos.

Para além do referido, foram desenhados para-choques para absorver a energia das colisões durante as partidas, evitando que os robôs sofressem danos (Fig. 56). Estes para-choques tinham a função adicional de facilitar a condução da bola, a partir de um encaixe dimensionado à medida da bola utilizada.

A componente da programação não foi abordada nesta atividade, pois cada grupo tinha apenas 45 min com o kit. Procurou-se cativar o interesse dos alunos recorrendo à competitividade inerente a qualquer jogo de futebol.

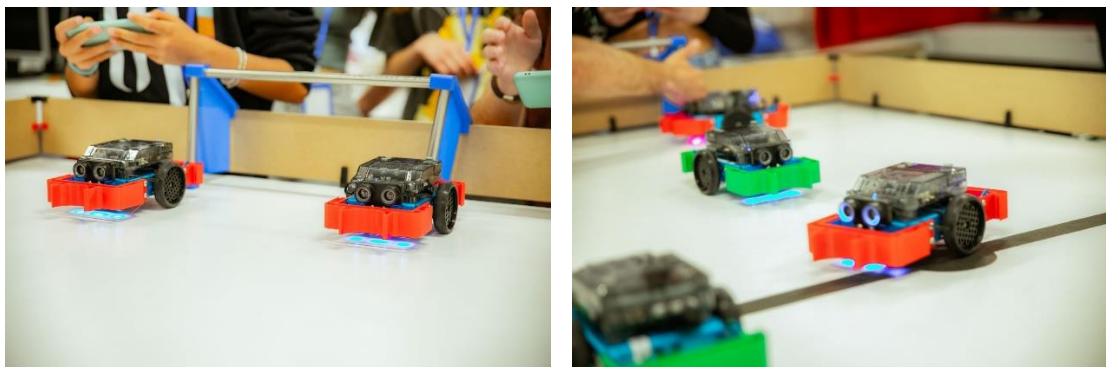


Fig. 56. Fotografias capturadas durante a realização do Futebot, torneio de futebol com os mBots. Destaque para as balizas e para-choques desenvolvidos para esta atividade recorrendo ao fabrico aditivo.

5.3. Protocolos desenvolvidos

O conjunto de trabalhos desenvolvidos, que inclui desde a experiência adquirida na formação com a Escola Profissional, o contacto com os alunos na Universidade de Inverno STEAM até às atividades no âmbito das unidades curriculares de Robótica, Sistemas Robotizados e Robótica Industrial, culminaram num objetivo principal. Esse objetivo consiste em desenvolver uma metodologia que desperte o interesse dos alunos pela robótica e que contribua para o desenvolvimento de competências como o pensamento crítico e o pensamento computacional, cruciais no mundo da Indústria.

Neste subcapítulo, serão explorados os protocolos desenvolvidos com base numa abordagem metodológica que promove uma aprendizagem sequencial. Esta abordagem estimula o pensamento dos alunos, concedendo-lhes a liberdade de construir um caminho para atingir um objetivo específico. Em cada um dos protocolos, o objetivo almejado é claramente definido, incentivando os alunos a aplicarem os seus conhecimentos e habilidades de forma estruturada e progressiva.

Recomenda-se em todas as atividades que seja utilizada a mesa de trabalho multiusos, uma vez que esta foi dimensionada para permitir diferentes configurações de trabalho, proporcionando um ambiente seguro, que reduz o risco de acidentes e quedas que possam danificar os robôs. Alternativamente, propõe-se que as atividades sejam feitas num ambiente que não permita o risco de quedas (p. ex. no chão). É recomendado o uso de fita de papel sempre que necessário definir uma linha para seguir, uma vez que a sua remoção não deixa resíduos de cola na mesa.

A Tabela 2 contém o objetivo de cada um dos protocolos e os materiais necessários para a realização dos mesmos.

Tabela 2. Protocolos desenvolvidos e respetivos recursos para a sua realização.

Atividade	Detalhes	
Configuração inicial e primeiros passos	Linguagem de programação: Scratch	Material necessário: Computador, mBot2, sensor de ultrassons, mesa de trabalho multiusos.
	Objetivo: O documento serve de apoio à primeira atividade realizada com o kit mBot2. Explica como instalar a aplicação e configurar a conexão com o mBot2. Por fim, um breve tutorial de como criar um programa, movimentando o robô em função dos valores devolvidos pelo sensor ultrassom.	
Programação tangível	Linguagem de programação: Python	Material necessário: mBot2, mesa de trabalho multiusos.
	Objetivo: Proporcionar aos alunos mais jovens uma compreensão intuitiva e <i>hands-on</i> dos conceitos fundamentais de programação. Este protocolo descreve a preparação de uma atividade de programação tangível. Assim sendo, o documento auxilia o professor/formador responsável pela realização da atividade.	
Futebol	Linguagem de programação: N/A	Material necessário: Smartphone ou tablet, mBot2, para-choques, bola, mesa de trabalho multiusos.
	Objetivo: Este protocolo destina-se a facilitar uma atividade interativa, lúdica e cativante que não requer competências de programação, rápida de preparar e explicar. Ideal para visitas rápidas de alunos de todas as idades ao laboratório de robótica. O Futebol consiste num jogo onde os mBots são controlados através da aplicação Makeblock, são como jogadores num minijogo de futebol. O documento explica a configuração da atividade, a instalação e o uso da aplicação Makeblock, além das regras do jogo. O objetivo é proporcionar uma experiência divertida e envolvente que permite aos visitantes interagir diretamente com os robôs, despertando assim o interesse pela robótica através de uma atividade competitiva.	
Segue-linha	Linguagem de programação: Scratch	Material necessário: Computador, mBot2, fita de papel (preta), Sensor Quad RGB, mesa de trabalho multiusos.
	Objetivo: Este trabalho tem o objetivo de introduzir os alunos ao funcionamento do sensor seguidor de linha e de deteção de cor Quad RGB. É a continuação lógica da atividade de configuração inicial, assim sendo, todos os passos são descritos detalhadamente para melhor compreensão do aluno. Propõe-se a realização de um trajeto com auxílio do sensor seguidor de linha. Recomenda-se a utilização da pista e do sensor Quad RGB incluídos no kit do mBot2. Apresenta-se uma lista de passos para a preparação e realização da atividade.	

Tabela 3. Protocolos desenvolvidos e respetivos recursos para a sua realização (cont.).

Desvio de obstáculos	Linguagem de programação: Scratch	Material necessário: Computador, mBot Ultimate 2.0, fita de papel, Sensor seguidor de linha, Sensor de ultrassons, obstáculo, mesa de trabalho multiusos.
	Objetivo: Este trabalho tem o objetivo de introduzir os alunos ao funcionamento do sensor de ultrassons. Propõe-se aos alunos que coloquem o mBot2 a percorrer um trajeto, no qual será colocado aleatoriamente um obstáculo (ex: bloco de madeira). O robô terá de ultrapassar essa adversidade, desviando ou contornando o objeto.	
Transporte de objetos	Linguagem de programação: Scratch	Material necessário: Computador, mBot2, fita de papel (várias cores), Sensor Quad RGB, garra, esfera metálica, mesa de trabalho multiusos.
	Objetivo: O objetivo desta atividade é introduzir os alunos aos outputs do mBot2. Propõe-se aos alunos que transportem um objeto entre dois pontos de um trajeto previamente definido. Para isso, é necessário instalar uma garra desenvolvida no âmbito desta dissertação, e que faz uso de uma porta de output do mBot2 para alimentar um servomotor que a controla. O robô terá de agarrar o objeto num ponto de recolha e movê-lo até ao ponto de entrega.	
Torre de blocos	Linguagem de programação: C++	Material necessário: Computador, Ultimate 2.0, Sensor de infravermelhos Sensor de ultrassons, blocos, mesa de trabalho multiusos.
	Objetivo: O objetivo deste protocolo é utilizar o Ultimate 2.0, com a configuração "Robotic Arm Tank", a manipular objetos com precisão, construindo uma torre com pelo menos 3 blocos de altura. O percurso e o método da construção ficam ao critério do grupo desde que o objetivo proposto seja alcançado. Sugere-se que utilizem as ferramentas desenvolvidas nos protocolos segue-linha e desvio de obstáculos para conseguir localizar a posição dos blocos com precisão suficiente para os empilhar verticalmente.	
Labirinto	Linguagem de programação: Python	Material necessário: Computador, mBot2, Sensor Quad RGB, Sensor de ultrassons, mesa de trabalho multiusos.
	Objetivo: O objetivo deste trabalho é aplicar algoritmos de planeamento de caminho, lecionados na unidade curricular de robótica, para colocar o mBot2 a percorrer um labirinto aleatório. O labirinto na mesa de trabalho multiusos será montado com os suportes e paredes (placas MDF) detalhadas na secção 4.1. da presente dissertação. A estratégia utilizada fica ao critério de cada grupo.	
Condução autónoma	Linguagem de programação: Scratch	Material necessário: Computador, mBot2, semáforos, Smart Camera, mesa de trabalho multiusos.
	Objetivo: Este trabalho visa ser integrado com um dos projetos habitualmente realizados na unidade curricular de Automação Industrial, o projeto de um sistema de controlo de semáforos, a partir de um controlador lógico programável (PLC). Pretende-se a programação dos mBot2 para que dirijam de forma autónoma, utilizando ferramentas de visão computacional possibilitadas pela Smart Camera. Assim, será possível que os mBot2 se dirijam obedecendo à indicação luminosa dos semáforos programados, simulando veículos numa situação de tráfego normal.	

O protocolo “Configuração inicial e primeiros passos” pretende introduzir à utilização dos mBot, quer do ponto de vista do professor/formador que irá preparar uma atividade, que do ponto de vista de alunos que se pretendam introduzir à programação destes robôs.

O protocolo “Programação Tangível” distingue-se de todos os outros, pois tem um caráter educativo sem necessitar que os alunos tenham qualquer conhecimento de programação. Tem com público-alvo crianças que ainda não tenham sido introduzidas à lógica da programação, permitindo que a trajetória do mBot seja programada pelo clicar de setas pela ordem pretendida.

O protocolo “Futebot” pretende auxiliar numa montagem rápida de um jogo didático e divertido com o objetivo de apoio às atividades de divulgação da robótica e da área da engenharia da UBI, a visitantes externos de todas as idades.

Os protocolos “Segue-linha” e “Desvio de obstáculos” seguem-se ao protocolo “Configuração inicial e primeiros passos” por terem um caráter introdutório, contendo todos os passos necessários para a realização das atividades. O grau de complexidade para os restantes protocolos vai incrementando gradualmente, passando de uma fase em que a programação é feita em Scratch, para o código desenvolvido em Python e C++, e passando de um caráter introdutório da programação na plataforma mBot para a aplicação de conhecimento adquirido nas disciplinas da robótica.

Os protocolos expostos na Tabela 2 encontram-se em anexo a esta dissertação.

Os programas que se encontram nos protocolos, estão disponíveis em: <https://github.com/rodrigoantunes2000/mBlock-codes>

6. Resultados

O capítulo dos Resultados diz respeito à formação realizada com os alunos da Escola Profissional. São descritas também algumas observações sobre o comportamento dos alunos na realização do Futebol durante a Universidade de Inverno STEAM.

6.1. Introdução

Durante a formação foi avaliado o desempenho, bem como a evolução de todo o grupo com o objetivo de implementar melhorias na abordagem de ensino.

A construção dos instrumentos de avaliação é geralmente complexa, deve-se ter em conta a definição de objetivos, o inventário de recursos disponíveis, a identificação dos indivíduos, selecionar a amostra representativa, elaborar o esboço do questionário, realizar o pré-teste, elaborar o questionário definitivo, recolher as informações, codificar as respostas, e fazer a análise e tratamento dos dados e elaborar o relatório final (Vilelas, 2017; Rodrigues, 2020).

Uma vez que cada turma teve apenas 3 aulas, com duração de 2 horas cada uma, os instrumentos de avaliação não poderiam ser demasiado complexos.

O plano de investigação deste estudo é de cariz qualitativo uma vez que parte dos dados foi recolhida através da observação direta, registado através de notas descritivas e/ou documentos de observação. Para complementar as informações recolhidas pela observação, recorreu-se à análise das respostas dos estudantes a dois questionários anónimos, assim como perguntas realizadas diretamente durante as aulas.

Um questionário inicial que servia como um diagnóstico do trabalho desenvolvido pelos alunos na área da robótica ao longo do seu percurso académico. O último questionário era disponibilizado no final da última aula, com o objetivo do estudante se autoavaliar e avaliar a metodologia de ensino seguida durante a formação.

O processo de colocar perguntas aos alunos, designado de entrevista (Coutinho, 2019), é mais flexível que as outras técnicas, permite que os participantes aceitem falar sobre determinados assuntos com uma certa abertura, permitindo uma cobertura mais aprofundada.

6.2. Questionário 1

O primeiro questionário era composto por um conjunto de 8 perguntas de resposta afirmativa ou negativa (sim ou não):

Num total de 56 participantes, a Tabela 4 apresenta a percentagem de respostas afirmativas e negativas a cada uma das questões, permitindo assim uma melhor noção do conhecimento prévio adquirido pelos estudantes e ainda realizar pequenos ajustes ao programa lecionado nas aulas seguintes.

Tabela 4. Percentagem de respostas afirmativas e negativas do primeiro questionário colocado aos participantes da formação dirigida ao ensino da robótica por via de kits de robótica móvel (mBot 2).

Questões	Sim	Não
Já tiveste contacto com algum robô?	94,64%	5,36%
Já programaste um robô?	69,64%	30,36%
Já utilizaste uma linguagem de programação por blocos?	35,71%	64,29%
Já utilizaste algum tipo de sensor durante o teu percurso académico?	64,29%	35,71%
Conheces a linguagem de programação “Scratch”?	35,71%	64,29%
Já ouviste falar em variáveis booleanas?	3,57%	96,43%
Conheces o conceito de algoritmo?	17,86%	82,14%
Já trabalhaste com <i>encoder motors</i> ?	3,57%	96,43%

6.3. Questionário 2

O último questionário focava-se na avaliação do interesse e opinião dos participantes sobre as aulas.

O número de participantes oscilou ao longo das aulas. Uma das turmas não compareceu na última aula e daí o número de respostas ao último questionário ser 44.

A Tabela 5 apresenta a percentagem de respostas afirmativas e negativas a cada uma das questões, com a discussão de resultados efetuada no capítulo seguinte.

Tabela 5. Percentagem de respostas afirmativas e negativas do último questionário colocado aos participantes da formação dirigida ao ensino da robótica por via de kits de robótica móvel (mBot 2).

Questões	Sim	Não	Faltou
Sentiste dificuldade durante a formação?	23,21%	55,36%	21,43%
Gostaste de trabalhar em equipa?	76,79%	1,78%	21,43%
Sentiste maior facilidade na programação com o decorrer das aulas?	64,28%	14,29%	21,43%
Foi desafiante programar com mBlock?	69,64%	8,93%	21,43%
Os problemas propostos foram importantes para o desenvolvimento do raciocínio?	69,64%	8,93%	21,43%
O orientador apoiou os alunos para a obtenção de sucesso na resolução das atividades?	75,00%	3,57%	21,43%
Sentes que esta formação será útil para o meu futuro?	67,86%	10,71%	21,43%
Gostarias de continuar a programar mBots?	66,07%	12,5%	21,43%

6.4. Observações durante a realização do Futebot

Embora não existam resultados mensurados para o desempenho dos alunos nesta atividade, existem algumas observações que merecem destaque. Nomeadamente, o elevado interesse dos alunos durante a realização da atividade, a diversão visível nas expressões faciais e, por último, uma dificuldade que surgiu na generalidade: a dificuldade de se colocarem no ponto de vista do robô quando tentavam recuar no terreno.

6.5. Nota Conclusiva

Com base nos questionários aplicados antes e após a formação com os mBots 2, foi possível obter resultados qualitativos que fornecem uma visão abrangente da perceção dos alunos em relação à metodologia, ao material utilizado e ao trabalho desenvolvido em grupo. Esta secção é fundamental para compreender o impacto da abordagem adotada na formação e identificar possíveis melhorias ou áreas em destaque.

Os resultados do primeiro questionário podem indicar o nível de conhecimento e familiaridade dos participantes com o conteúdo abordado na formação, enquanto as respostas no questionário

pós-formação refletem o impacto da experiência vivida e da aprendizagem adquirida ao longo do curso.

No próximo capítulo, procede-se à análise e discussão detalhada desses resultados, sendo exploradas as respostas dos alunos em relação a diferentes aspetos da formação, incluindo a compreensão da metodologia, a perceção sobre a eficácia do material utilizado, a experiência de trabalhar em grupo entre outras questões relevantes.

7. Análise de Discussão de Resultados

7.1. Formação Escola Profissional

Tal como mencionado anteriormente, esta intervenção realizou-se com alunos do secundário de uma Escola Profissional.

Com base nos resultados do primeiro questionário (Tabela 4), retiramos que a maior parte dos alunos diz já ter tido contacto com robôs. Cerca de 70% diz já ter programado um robô, 64,3% já trabalhou com sensores ao longo do percurso académico. Porém poucos afirmam ter trabalhado com *encoder motors* (menos de 4%). Este último resultado revela que o trabalho realizado com robótica por parte dos participantes foi muito superficial, algo que se veio a confirmar também durante o decorrer das aulas.

Uma minoria dos alunos já tinha trabalhado com Scratch (35,7%), após diálogo com os mesmos foi perceptível que não abordaram estruturas de controlo como *loops* e condições (*if*) na realização de programas.

Apesar da recolha de informação no primeiro questionário ser de perguntas diretas afirmativas/negativas, a complementaridade do diálogo com os alunos permitiu compreender que os alunos estavam num nível muito próximo de conhecimento ao nível da programação.

A preocupação no segundo questionário não foi avaliar os alunos quanto ao conhecimento pois o tempo de formação dificilmente permitiria uma avaliação contínua do progresso. O objetivo foi, unicamente, perceber de que forma a metodologia utilizada inspirou os alunos na aprendizagem da robótica e se existe interesse em explorar esta vertente no futuro.

A maioria dos alunos (66,1%) afirmou que gostaria de continuar a programar mBots 2 se lhe fosse dada a oportunidade. Até os professores da escola profissional mostraram interesse na aquisição dos kits de robótica móvel. Relativamente à dificuldade, 23,2% admite ter sentido dificuldade na realização das atividades, mas 64,3% reconhece que as dificuldades foram diminuindo com o decorrer das aulas.

Em relação à metodologia, 69,6% acredita que os problemas propostos foram importantes para o desenvolvimento do raciocínio e 75% concorda que o apoio fornecido durante as aulas contribuiu para o sucesso na resolução das atividades, estimulando a participação, dando tempo aos alunos para refletirem sobre os problemas.

Esta compilação de resultados termina com uma grande maioria dos participantes (67,9%) reconhecer que a formação foi útil para o futuro, seja ele acadêmico ou profissional.

Um dos grandes objetivos da formação, para além de promover o pensamento crítico e o interesse na robótica, era fomentar o trabalho em equipa. Esse objetivo foi relativamente bem-sucedido, com 76,8% dos alunos a assumir que gostou de resolver os desafios em grupo.

7.2. Nota Conclusiva

Os resultados obtidos dos questionários revelam que os alunos da Escola Profissional possuíam algum nível de familiaridade com robótica, principalmente em termos de contato com robôs e programação básica. No entanto, o conhecimento em programação e utilização de componentes mais avançados, como *encoder motors*, era limitado. A metodologia utilizada na formação demonstrou ter um impacto positivo nos alunos, despertando interesse e inspiração na aprendizagem da robótica. A maioria dos alunos expressou o desejo de continuar a programar mBots 2 e reconheceu a importância dos problemas propostos e do apoio fornecido durante as aulas. Além disso, a formação foi considerada útil para o futuro acadêmico e profissional dos participantes. O trabalho em equipa também foi valorizado, com a maioria dos alunos expressando apreço pela resolução de desafios em grupo.

No capítulo seguinte são expostas as conclusões do trabalho de dissertação, focando os pontos fortes da abordagem como despertar o interesse dos alunos, promover o pensamento crítico e desenvolver competências de resolução de problemas. Enumera as limitações identificadas, como o conhecimento prévio superficial dos alunos, a necessidade de aprofundar o ensino de programação e a utilização de componentes mais avançados.

8. Conclusões

8.1. Conclusões gerais

Muitos alunos têm experiência como utilizadores na interação com novas tecnologias. A tecnologia é capaz de fomentar a descoberta e a aprendizagem em ambiente de sala de aula e também fora da mesma.

Existe um conjunto de competências do pensamento computacional que podem beneficiar desta aproximação com a tecnologia e a finalidade de uma disciplina de robótica é capacitar os estudantes para a resolução de problemas.

Nas aulas lecionadas procurou-se não seguir o ensino tradicional, onde o professor é visto como o detentor do conhecimento e o aluno o recetor, utilizava-se o ensino expositivo e a comunicação unidirecional, pelo que a interação com o aluno era mais reduzida (Rodrigues, 2020). Nesta formação existiu constante partilha e esclarecimento de dúvidas, para que o ritmo da aula fosse dinâmico e não permitindo que os alunos dispersassem das tarefas que lhes eram propostas.

Durante as aulas, destaca-se a importância da elaboração de vários problemas, de grau de complexidade crescente. Na resolução do problema final, os alunos aplicaram os conhecimentos aprendidos durante a formação.

A escolha da robótica educativa foi efetuada para reforçar as competências, promover a interdisciplinaridade, promover o pensamento computacional, o raciocínio, criatividade, persistência e cativar os alunos a desenvolver outras competências necessárias no desenvolvimento da engenharia.

Nas primeiras duas aulas, de uma forma geral, existiu alguma dificuldade na implementação de “loops”, bem como na conjugação do sensor de linha com o sensor ultrassom, pois os alunos tinham que priorizar a informação recebida por um dos sensores para que o robô fosse capaz de tomar decisões sem entrar em conflito com outras instruções.

Na última aula, talvez devido à complexidade do desafio final, a competitividade foi desaparecendo e existiu um ambiente de cooperação entre alguns grupos com o objetivo comum de encontrar uma estratégia capaz de percorrer todo o labirinto independentemente da sua configuração.

Apesar de alguns grupos terem conseguido colocar o robô a percorrer o labirinto, a estratégia definida não era infalível. Existiam configurações impossíveis de resolver pelo que os alunos necessitariam de mais tempo e aulas para atingir uma solução capaz de resolver qualquer cenário.

Com a observação anterior, destaca-se a importância do tempo neste tipo de atividades. Os alunos necessitam de um número superior de aulas para adquirir o conhecimento de forma mais gradual, já que a programação é uma disciplina encarada com grande dificuldade não só pelos alunos do ensino secundário, mas também do ensino superior.

8.2. Sugestões de trabalhos futuros

Na realização de trabalhos futuros, para além do aumento do número de aulas, seria importante implementar outros instrumentos de recolha de dados que permitissem, efetivamente, perceber os efeitos da robótica educativa, pois as respostas ao segundo questionário não permitem retirar conclusões relativamente ao progresso dos alunos ao nível do pensamento crítico ou capacidade de raciocínio, pois segue a linha de raciocínio do primeiro questionário, apenas existem perguntas de resposta afirmativa/negativa.

Relativamente aos protocolos desenvolvidos, seria importante testá-los em contexto de sala de aula e compreender quais os efeitos que têm sobre os alunos.

Não houve oportunidade de testar o protocolo que integra o projeto dos semáforos desenvolvido na unidade curricular de Automação Industrial, pois a aquisição das *Smart Cameras* não foi possível durante a realização da dissertação.

Referências Bibliográficas

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661-670.

Automation. (2022). *Automation*. Obtido de IFR Presents World Robotics Report 2022: <https://www.automation.com/en-us/articles/october-2022/ifr-presents-world-robotics-report-2022>

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 58(3), 978-988.

Bers, M., Seddighin, S., & Sullivan, A. (2013). Ready for robotics: Bringing together the T and E of STEM in early childhood teacher education. *Journal of Technology and Teacher Education*, 21(3), 355-377.

Çakır, R., Korkmaz, Ö., İdil, Ö., & Erdoğmuş, F. U. (2021). The effect of robotic coding education on preschoolers' problem solving and creative thinking skills. *Thinking Skills and Creativity*, 40, 100812.

Candelas, F. A., Puente, S. T., & Torres, F. (2016, September). Competition benchmarking to design and program mobile robots. In *2016 IEEE Conference on Control Applications (CCA)* (pp. 839-844). IEEE.

Chambers, J. M., Carbonaro, M., & Murray, H. (2008). Developing conceptual understanding of mechanical advantage through the use of Lego robotic technology. *Australasian Journal of Educational Technology*, 24(4).

Coll, C., & Solé, I. (2001). Os professores e a concepção construtivista. C. Coll, E. Martín, T. Mauri, M. Miras, J. Onrubia, I. Solé, & A. Zabala, *O construtivismo na sala de aula: Novas perspectivas para a ação pedagógica*, 8-27.

Computer Science Teachers Association (CSTA) Standards Task Force. (2011). *CSTA K-12 computer science standards*. http://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/Docs/Standards/CSTA_K-12_CSS.pdf

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33-39.

- Diago, P. D., González-Calero, J. A., & Yáñez, D. F. (2022). Exploring the development of mental rotation and computational skills in elementary students through educational robotics. *International Journal of Child-Computer Interaction*, *32*, 100388.
- Frank, M. C., Wysk, R. A., & Joshi, S. B. (2004). Rapid planning for CNC milling—A new approach for rapid prototyping. *Journal of Manufacturing Systems*, *23*(3), 242-255.
- Hamilton, M., Clarke-Midura, J., Shumway, J. F., & Lee, V. R. (2020). An emerging technology report on computational toys in early childhood. *Technology, Knowledge and Learning*, *25*(1), 213-224.
- Janzen, D. S., & Saiedian, H. (2006). Test-driven learning: intrinsic integration of testing into the CS/SE curriculum. *Acm Sigcse Bulletin*, *38*(1), 254-258.
- Javaid, M., Haleem, A., Singh, R. P., & Suman, R. (2021). Substantial capabilities of robotics in enhancing industry 4.0 implementation. *Cognitive Robotics*, *1*, 58-75.
- Johnson, J. (2003). Children, robotics, and education. *Artificial Life and Robotics*, *7*, 16-21.
- Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The Effect of a Classroom- Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood. *Early Childhood Education Journal*, *41*(4), 245–255.
- Khan, M., Haleem, A., & Javaid, M. (2023). Changes and improvements in Industry 5.0: A strategic approach to overcome the challenges of Industry 4.0. *Green Technologies and Sustainability*, *1*(2), 100020.
- Kiel, D., Müller, J. M., Arnold, C., & Voigt, K. I. (2017). Sustainable industrial value creation: Benefits and challenges of industry 4.0. *International journal of innovation management*, *21*(08), 1740015.
- Lakoff, G., & Núñez, R. (2000). *Where mathematics comes from* (Vol. 6). New York: Basic Books.
- Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O. S., Hubert, T., & Almughyirah, S. (2016). Using robotics and game design to enhance children’s self-efficacy, STEM attitudes, and computational thinking skills. *Journal of Science Education and Technology*, *25*, 860-876.
- Levichev, N., Vetrano, M. R., & Duflou, J. R. (2023). Melt flow and cutting front evolution during laser cutting with dynamic beam shaping. *Optics and Lasers in Engineering*, *161*, 107333.
- Makeblock. (2022). EN mBot2 Getting Started Activities V1.1.
- Makeblock. (2022). *mBlock Block-Based Editor- About mBlock5- What is mBlock 5 Block-Based Editor*. Obtido de Makeblock: <https://education.makeblock.com/help/about-mblock-5-block-based-editor/>

- Makeblock. (2 de May de 2022). *mBlock Block-Based Editor- Program a Device(Details)-Ultimate 2.0*. Obtido de Makeblock Education: <https://education.makeblock.com/help/mblock-block-based-mblock-block-based/>
- Makeblock. (2022). *mBlock Python Editor- About mBlock 5 Python Editor- What is mBlock-Python Editor*. Obtido de Makeblock education: <https://education.makeblock.com/help/mblock-python-editor-what-is-mblock-python-editor/>
- Makeblock. (2022). *Smart Camera*. Obtido de Makeblock education: <https://education.makeblock.com/help/mbuild-smart-camera/>
- Makeblock. (2022). *Ultimate_senior*. Obtido de Makeblock-help-center-zh: https://www.yuque.com/makeblock-help-center-zh/ultimate_senior/parameter
- Mead, R. A., Thomas, S. L., & Weinberg, J. B. (2012). From grade school to grad school: An integrated STEM pipeline model through robotics. In *Robots in K-12 education: A new technology for learning* (pp. 302-325). IGI Global.
- National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking*. National Academies Press.
- Ngo, T. D., Kashani, A., Imbalzano, G., Nguyen, K. T., & Hui, D. (2018). Additive manufacturing (3D printing): A review of materials, methods, applications and challenges. *Composites Part B: Engineering*, 143, 172-196.
- Onwubolu, G. C. (2017). *Introduction to SolidWorks: A comprehensive guide with applications in 3D printing*. CRC Press.
- Papert, S. (1980). *Mindstorms* Harvester Press.
- Pattis, R. E. (1990). A philosophy and example of CS-1 programming projects. *ACM SIGCSE Bulletin*, 22(1), 34-39.
- Pisarov, J., & Mester, G. (2019, December). Programming the mbot robot in school. In *Proceedings of the International Conference and Workshop Mechatronics in Practice and Education, MechEdu* (pp. 45-48).
- Priberam Informática, S. A. (n.d.). *Página principal*. Dicionário Priberam. Retrieved February 27, 2023, from <https://dicionario.priberam.org/>
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education*, 13(2), 137-172.
- Rodrigues, A. M. G. (2020). *Aplicação da robótica na resolução de problemas: Reflexões para a aprendizagem inicial de programação no ensino básico* (Doctoral dissertation, Universidade de Lisboa (Portugal)).

- Roumani, H. (2006). Practice what you preach: full separation of concerns in CS1/CS2. In SIGCSE Bulletin, 38(1), 58-64
- Schina, D., Esteve-Gonzalez, V., & Usart, M. (2021). Teachers' perceptions of bee-bot robotic toy and their ability to integrate it in their teaching. In *Robotics in Education: Methodologies and Technologies* (pp. 121-132). Springer International Publishing.
- Shamim, S., Cang, S., Yu, H., & Li, Y. (2016, July). Management approaches for Industry 4.0: A human resource management perspective. In *2016 IEEE congress on evolutionary computation (CEC)* (pp. 5309-5316). IEEE.
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational research review*, 22, 142-158.
- Sobota, J., Pišl, R., Balda, P., & Schlegel, M. (2013). Raspberry Pi and Arduino boards in control education. *IFAC Proceedings Volumes*, 46(17), 7-12.
- Solomon, C., Harvey, B., Kahn, K., Lieberman, H., Miller, M. L., Minsky, M., ... & Silverman, B. (2020). History of logo. *Proceedings of the ACM on Programming Languages*, 4(HOPL), 1-66.
- Tedre, M., & Denning, P. J. (2016, November). The long quest for computational thinking. In *Proceedings of the 16th Koli Calling international conference on computing education research* (pp. 120-129).
- Veiros, A. F. R. (2020). *Sistema robótico terrestre para apoio a atividades de manutenção de solo em pomares de prunóideas* (Doctoral dissertation).
- Vilelas, J. (2017). *Investigação – O Processo de Construção do Conhecimento* (2.^a Edição). Edições Sílabo.
- Voštinár, P. (2020, September). Using mBot robots for the motivation of studying computer science. In *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)* (pp. 653-657). IEEE.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of science education and technology*, 25, 127-147.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Xia, L., & Zhong, B. (2018). A systematic review on teaching and learning robotics content knowledge in K-12. *Computers & Education*, 127, 267-282.

Anexos



Departamento de
Engenharia Eletromecânica

Protocolo Laboratorial

Robótica, Sistemas Robotizados, Robótica Industrial

Configuração inicial e primeiros passos

Departamento de Engenharia Eletromecânica
Universidade da Beira Interior

Covilhã e UBI, junho de 2023

Objetivos do protocolo

Este documento serve de apoio à configuração inicial e primeiros passos com o kit mBot2.

A primeira parte diz respeito à instalação da aplicação e configurações da conexão com o mBot2.

Na segunda parte, um breve tutorial de como criar um programa, movimentando o robô em função dos valores devolvidos pelo sensor de ultrassons.

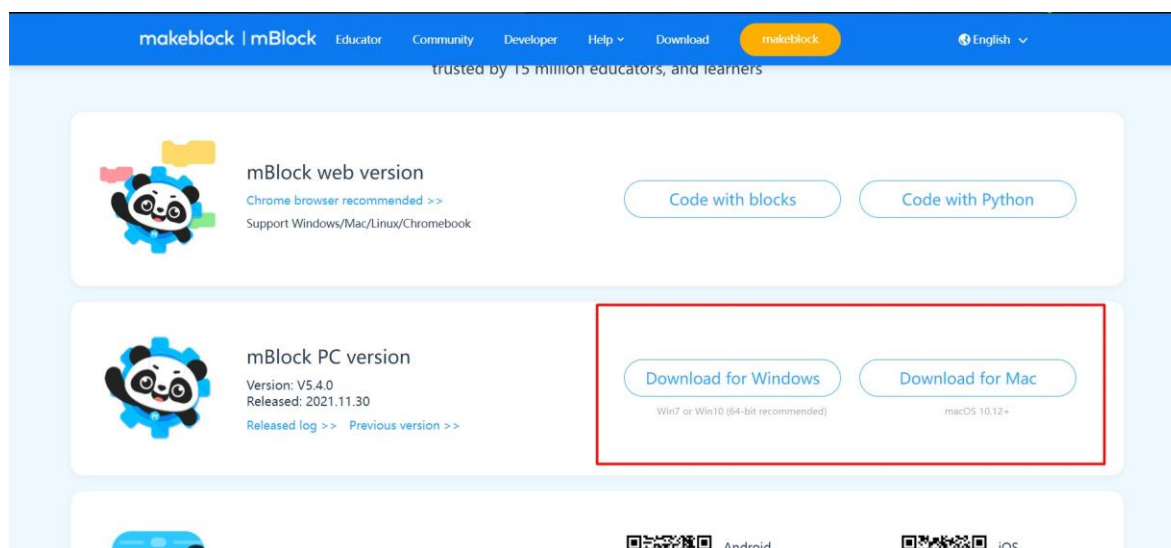
Recursos Necessários

1. Kit mBot2, incluindo cabos de conexão, sensor de ultrassons e sensor Quad RGB
2. Carregador Type-C (5V).
Nota: A bateria do CyberPi é alimentada enquanto conectado ao computador.
3. Computador com mBlock 5 instalado (na lista de passos está explícito como fazê-lo)

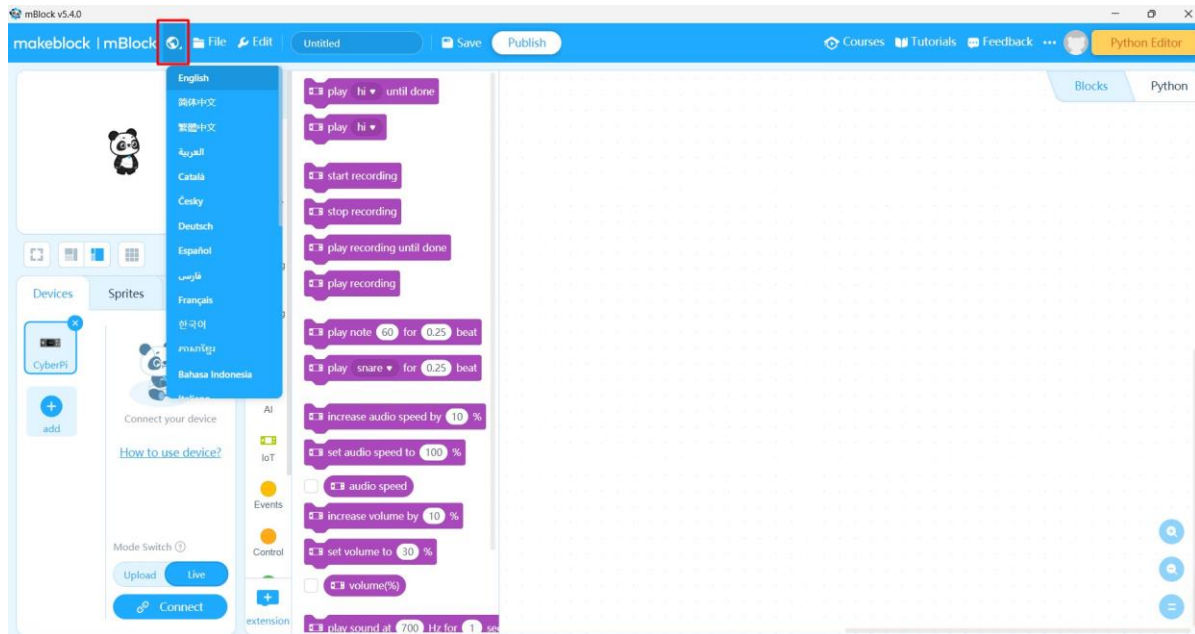
Lista de passos a seguir para instalar mBlock 5

1. Descarregue e instale o mBlock 5 no seu computador. Aceda à página:

<https://www.mblock.cc/en-us/download/>

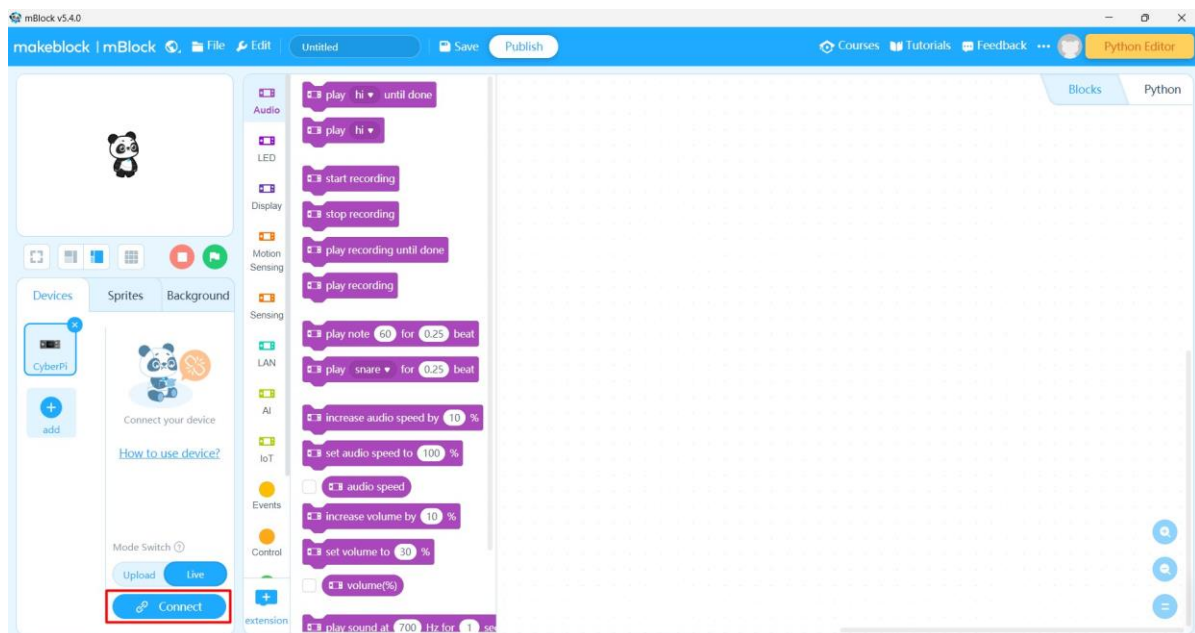


2. É aconselhado alterar o idioma para inglês, pois os termos utilizados nos blocos de programação assemelham-se a linguagens como C++, Python, Java, etc.

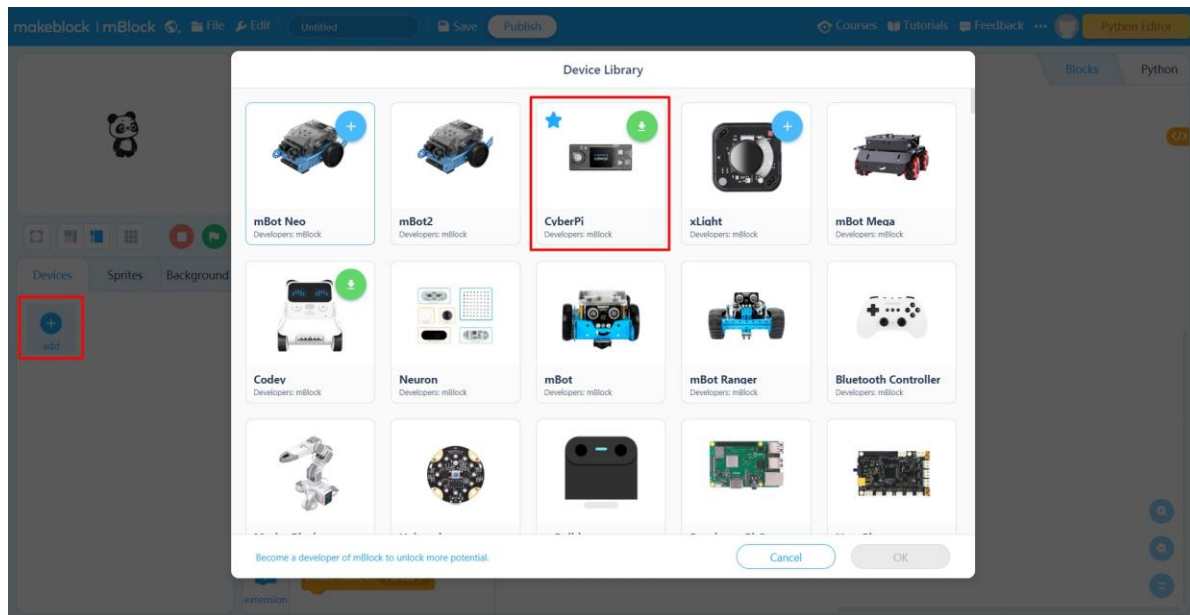


3. Conecte o mBot2 ao seu computador usando um cabo USB.

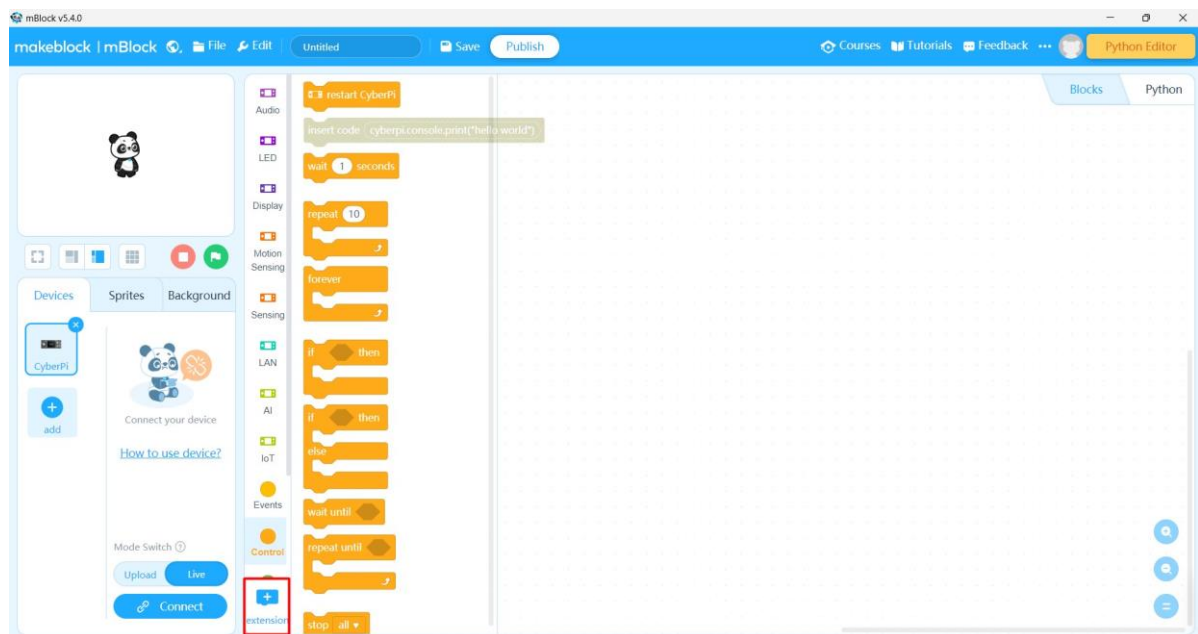
4. Abra o mBlock 5 e clique em "Connect" para conectar o mBot2 ao mBlock 5.



5. Adicione a extensão CyberPi ao seu projeto clicando em "add" na barra de ferramentas do mBlock 5 e pesquisando por "CyberPi".



6. Adicione outras extensões necessárias ao seu projeto, como sensores ou motores. Para a primeira atividade, será apenas necessário adicionar mBot2 shield, Quad RGB Sensor, e o Ultrasonic Sensor 2.















mBlock v5.4.0

Extension center

Back Create Extension

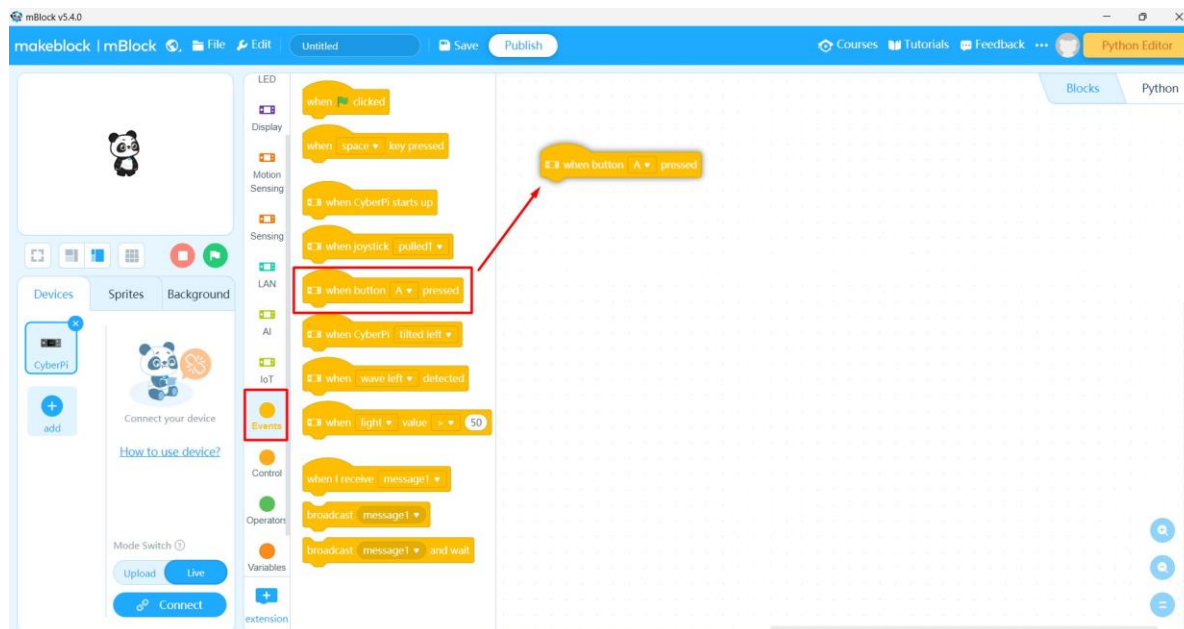
Search Sprite Extensions Device Extensions

CyberPi

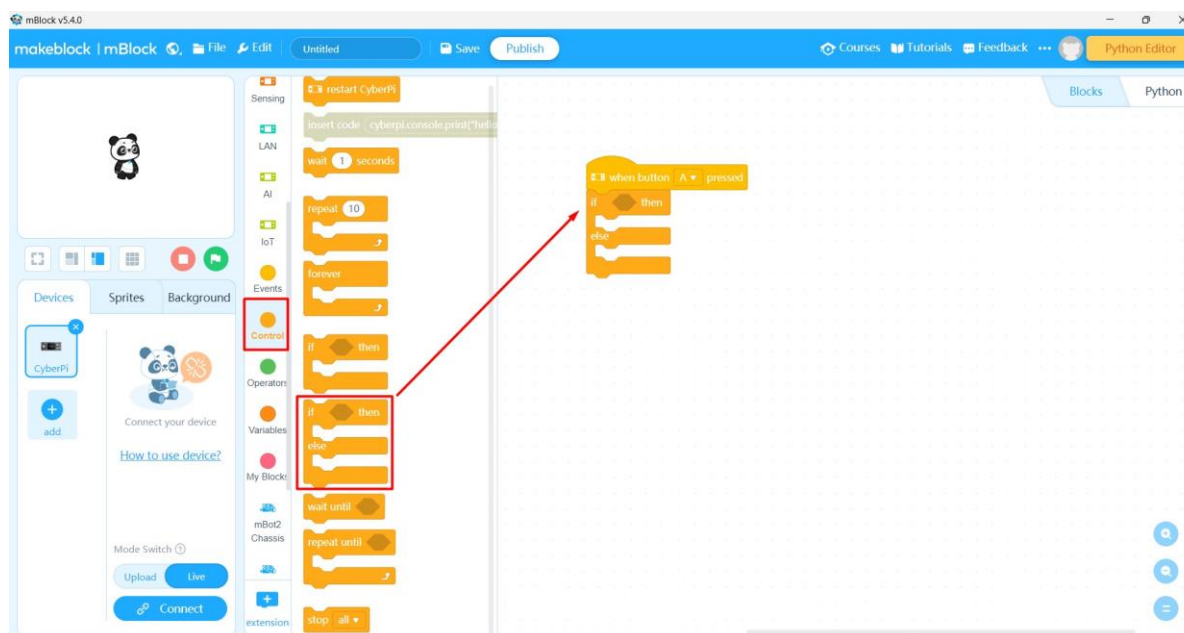
 <p>Pocket Shield By mBlock official Pocket Shield</p> <p>+ Add</p>	 <p>mBot2 shield By mBlock official More</p> <p>+ Add</p>	 <p>Ultrasonic Sensor 2 By mBlock official More</p> <p>+ Add</p>	 <p>Quad RGB Sensor(beta) By mBlock official</p> <p>+ Add</p>	 <p>Science Sensor By mBlock official An all-in-one sensor which integrates multiple electronic components for several purposes More</p> <p>+ Add</p>	 <p>Smart Camera By mBlock official Computer Vision</p> <p>+ Add</p>
 <p>Data Chart By mBlock official Use this extension to visualize your data. A chart speaks louder than</p>	 <p>Bluetooth controller By mBlock official Use the Bluetooth controller to make your robot do whatever you would</p>	 <p>Display+ By mBlock official Create games and UI elements on the display. More</p>	 <p>Slider By mBlock official More</p>	 <p>Humiture Sensor By mBlock official More</p>	 <p>LED Driver By mBlock official More</p>

Como criar um programa e implementá-lo no mBot2

1. Arraste e solte os blocos de programação para criar um programa. Para dar início a um programa é necessário arrastar um bloco do conjunto "Events".

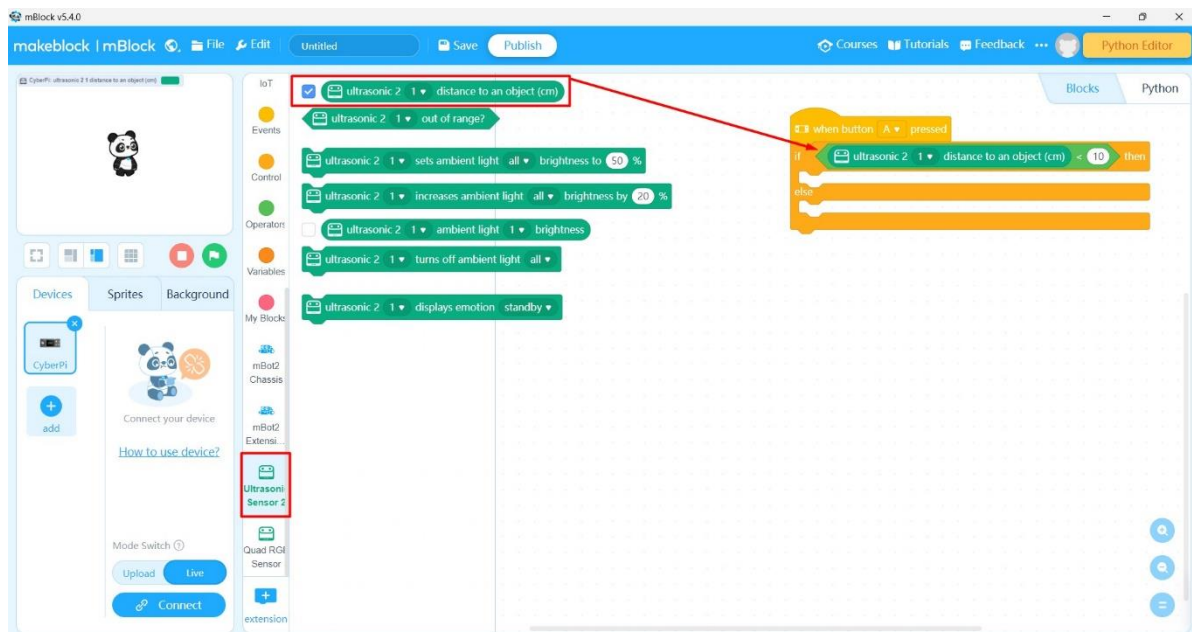
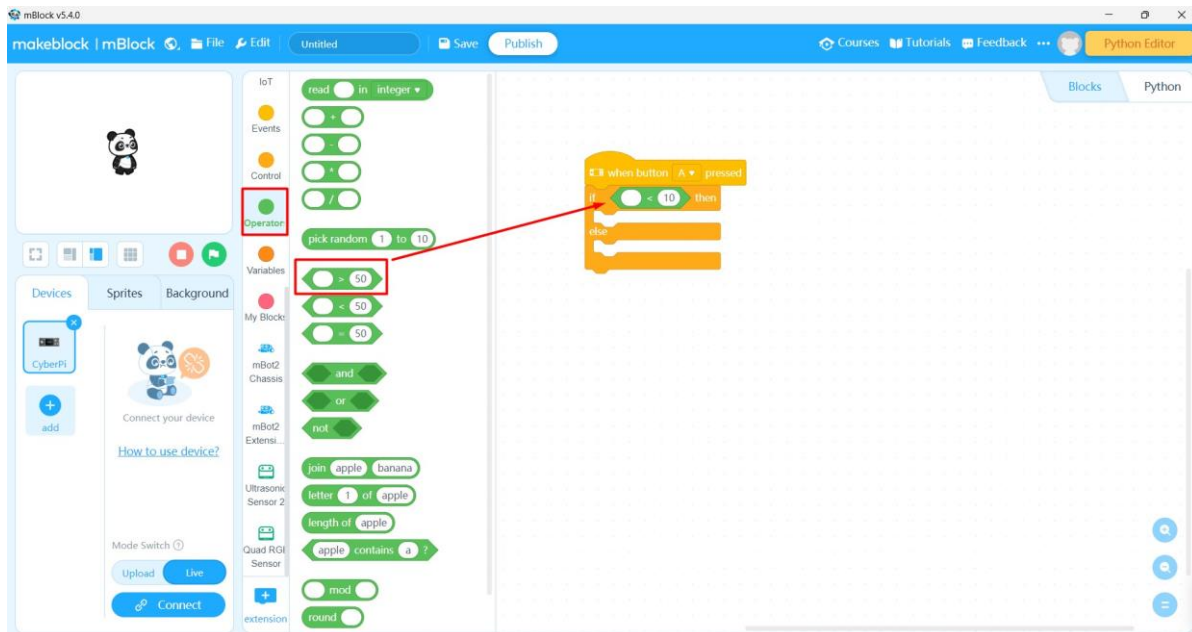


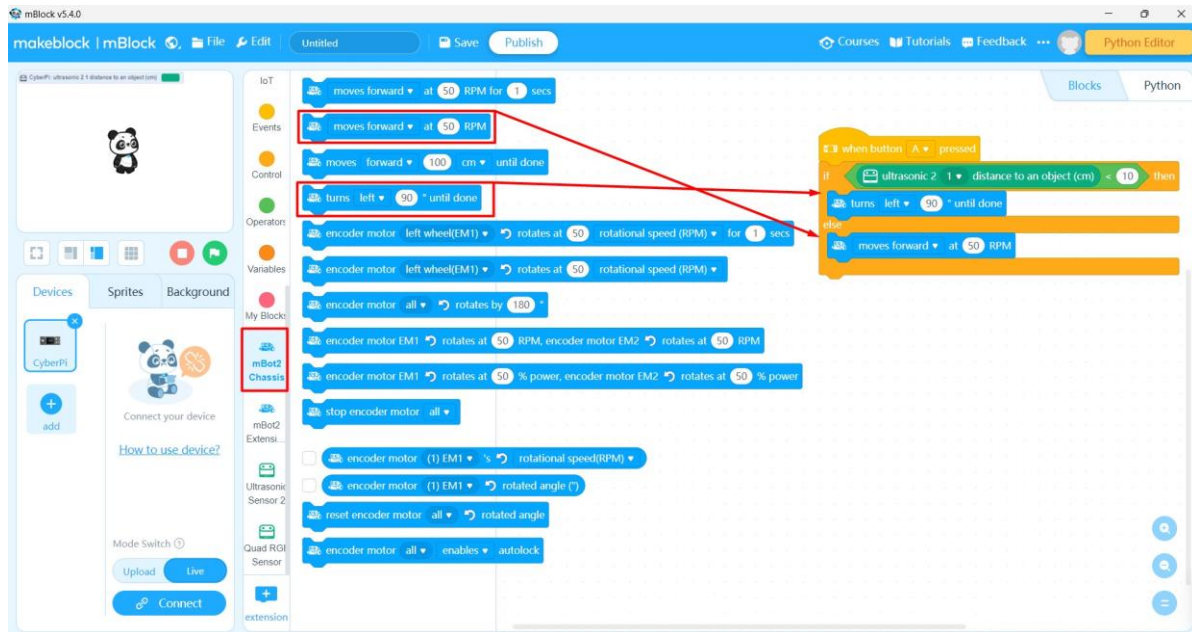
2. Os blocos relativos a expressões de condições e ciclos, encontram-se no conjunto "Control".



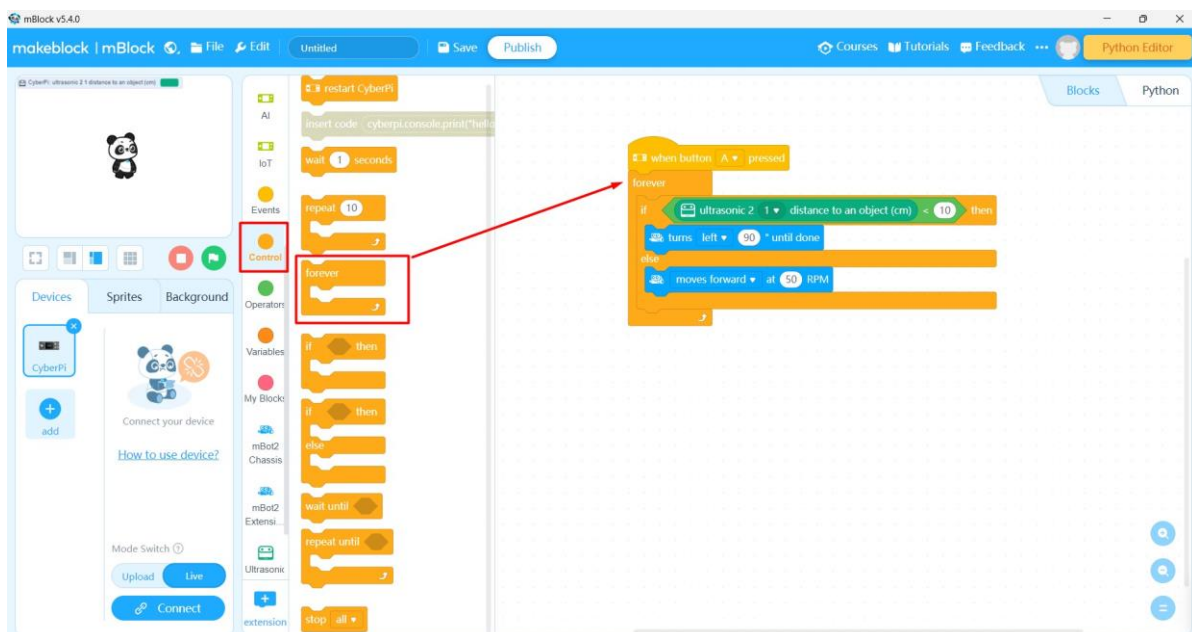
3. Para adicionar ordens para manipular o movimento do robô em função dos valores transmitidos pelos sensores, recorre-se aos blocos disponibilizados pelas extensões do mBot2 shield, Quad RGB Sensor e Ultrasonic Sensor 2.

Se pretende realizar operações lógicas ou comparar variáveis, arrasta-se os blocos do conjunto "Operators".

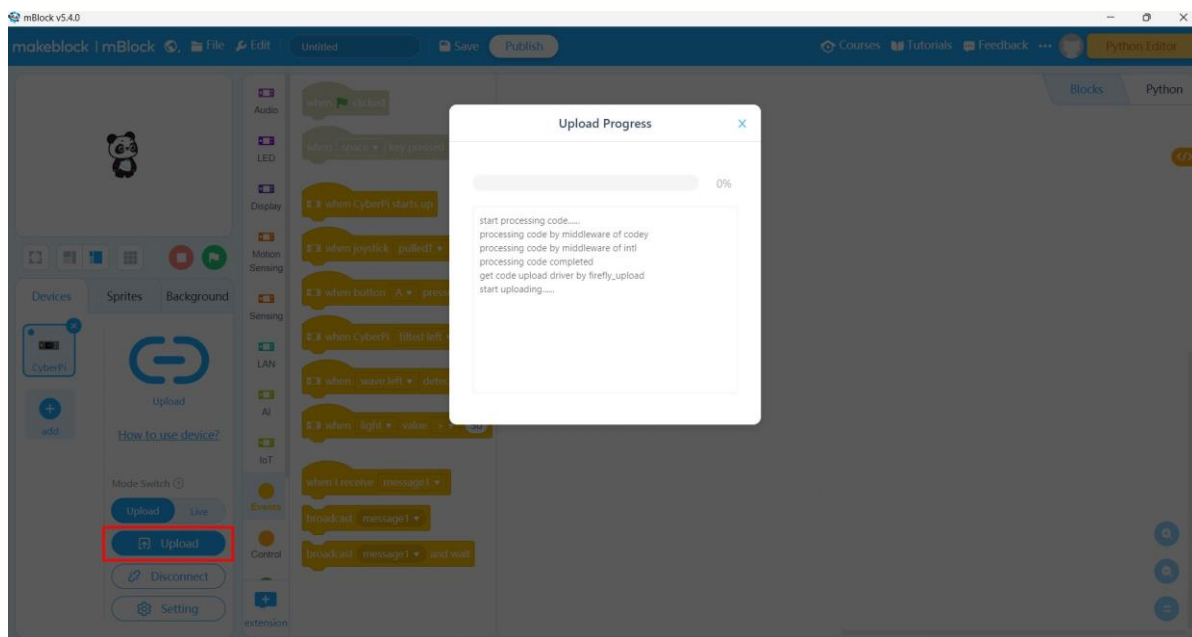
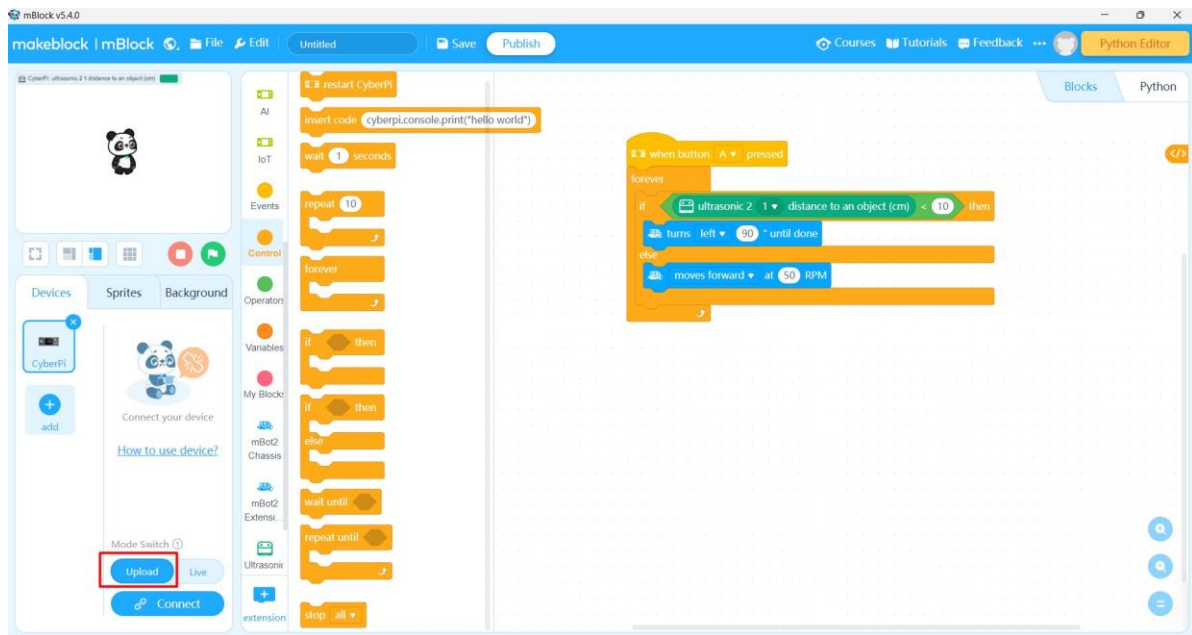




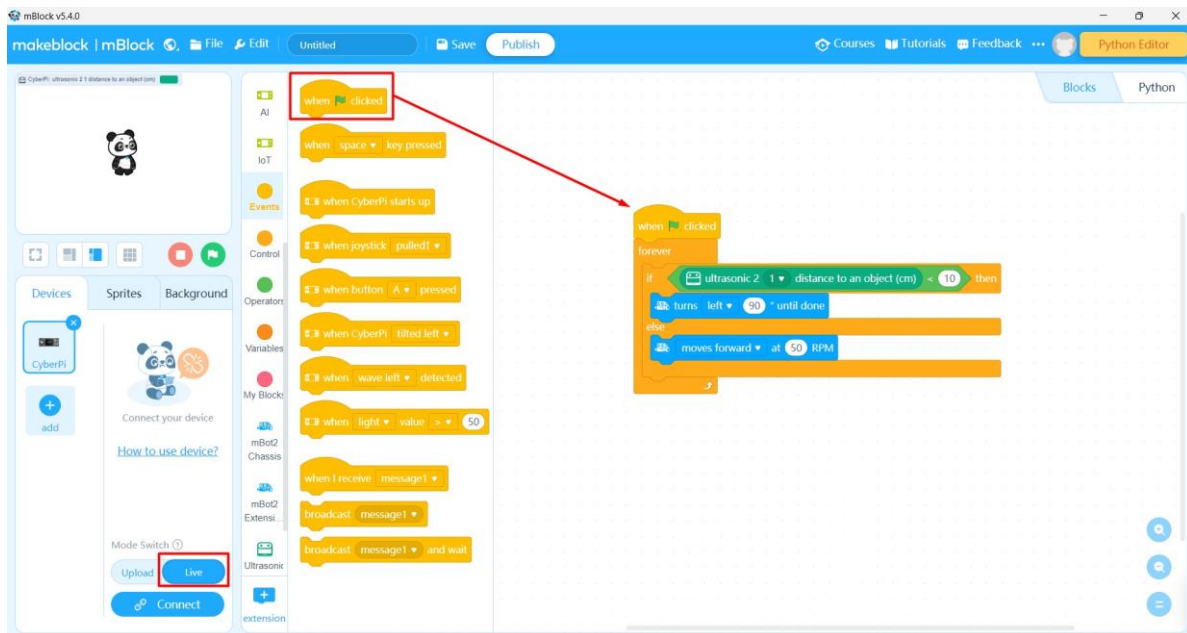
No exemplo apresentado, o mBot2 irá executar o programa quando o Botão A do CyberPi for pressionado (Botão com formato de uma seta). O robô irá verificar se existe algum objeto a uma distância inferior a 10 cm, caso se verifique ele realiza uma rotação de 90 graus no sentido positivo, de outra forma irá movimentar-se para a frente com uma velocidade de 50 RPM. Para que o robô esteja sempre a verificar esta condição é necessário acrescentar um ciclo:



4. Quando estiver pronto para executar o programa, clique em "Upload" para carregar o programa no mBot2. Nota: Para carregar o programa é necessário manter o robô ligado.

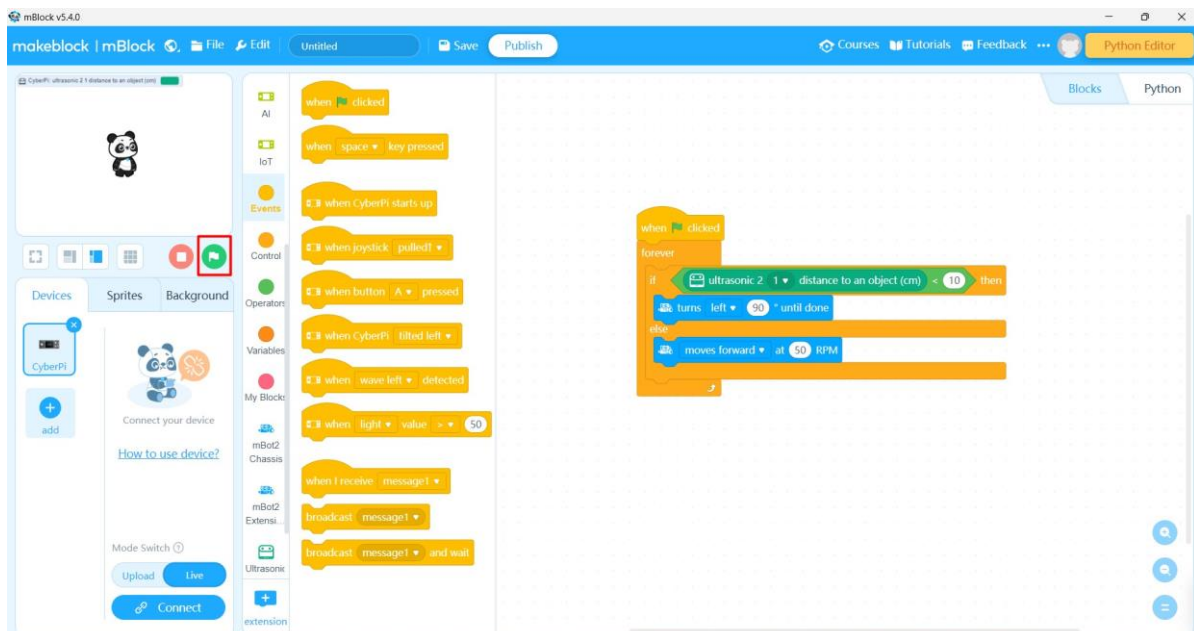


5. Caso pretenda testar o programa de forma direta sem carregar para o robô, pode optar pela opção "Live". Alterando apenas o evento que lhe dá início.



6. Por fim, basta clicar na bandeira verde para o robô executar o programa na opção "Live".

Nota: Antes de clicar na bandeira verde, verifique que existe espaço para o robô se movimentar sem criar tensão no cabo que o conecta ao computador.





Departamento de
Engenharia Eletromecânica

Protocolo Laboratorial

Robótica, Sistemas Robotizados, Robótica Industrial

Programação tangível

Departamento de Engenharia Eletromecânica
Universidade da Beira Interior

Covilhã e UBI, junho de 2023

Objetivos do protocolo

Este protocolo destina-se à preparação de uma atividade de programação tangível para utilizadores mais jovens. Assim sendo, o presente documento auxilia o professor/formador responsável pela realização da atividade.

A programação tangível permite ao utilizador pressionar as setas que se encontram à direita do display do CyberPi, criando uma sequência de movimentos pela ordem em que as setas são pressionadas.

O cenário construído na mesa de trabalho multiusos deve conter um trajeto com vários checkpoints (ex: criar uma cidade com uma escola, supermercado, igreja, casas, etc). Posteriormente criar diferentes itinerários que serão realizados segundo uma sequência de movimentos do mBot2. A sequência será realizada pelos alunos.

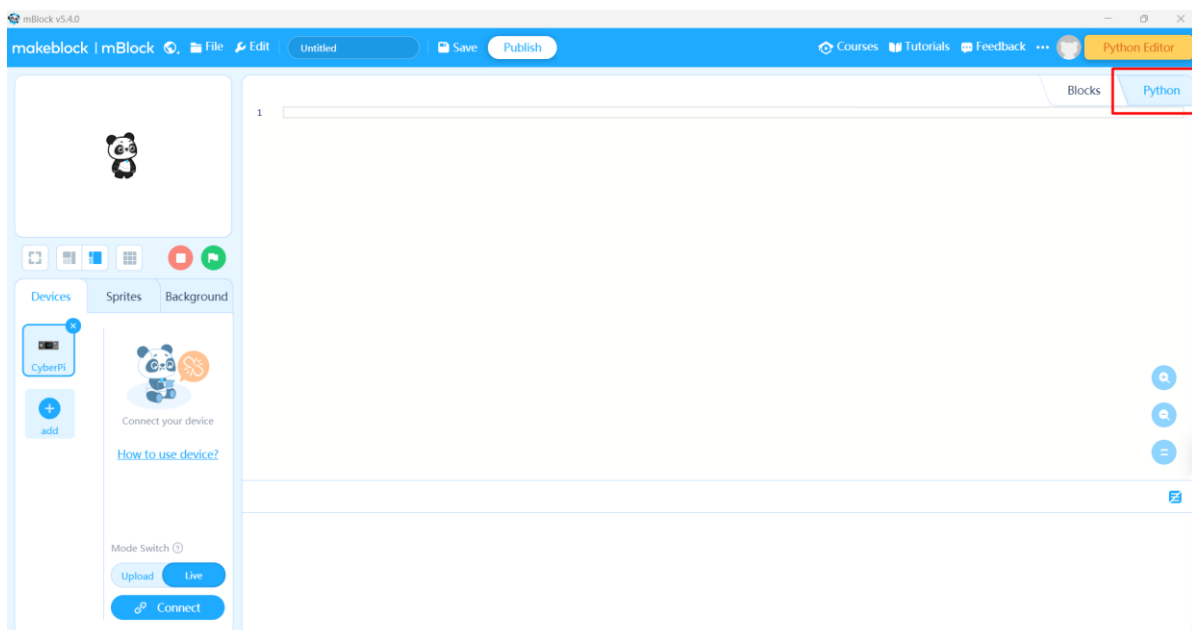
O script do programa deve ser escrito previamente pelo professor/formador, não sendo objetivo deste trabalho que os utilizadores tenham contacto direto com o código. Seguem-se as instruções para a construção do mesmo.

Recursos necessários

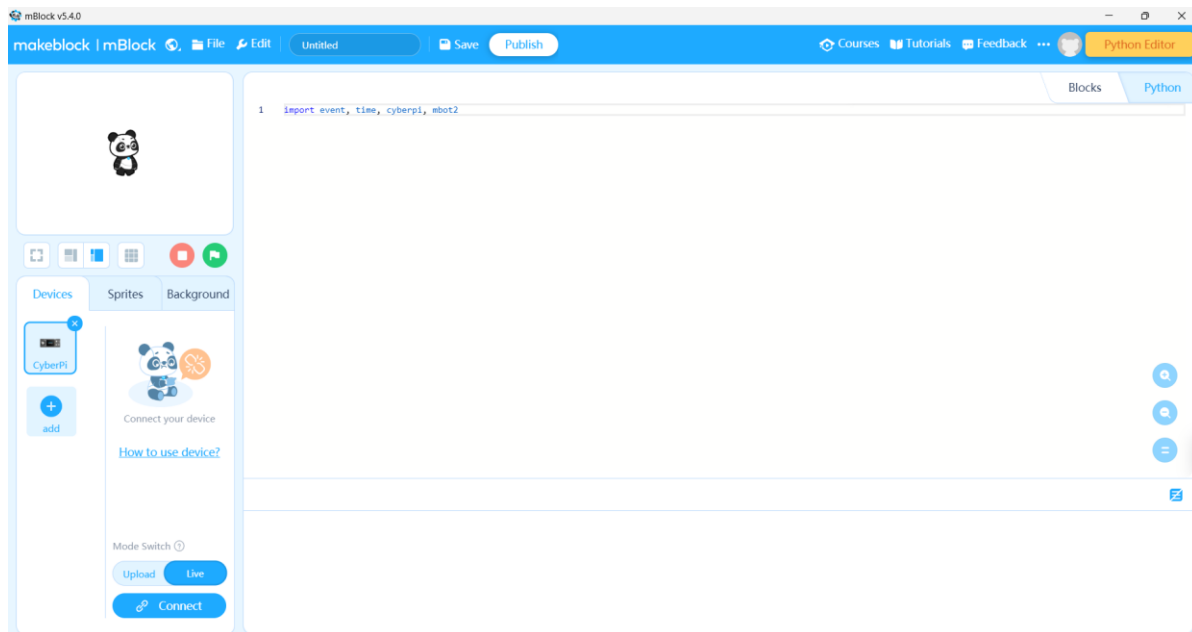
1. Kit mBot2, incluindo cabos de conexão.
2. Carregador Type-C (5V).
Nota: A bateria do CyberPi é alimentada enquanto conectado ao computador.
3. Computador com mBlock 5 instalado (Ver Protocolo “Configuração inicial e primeiros passos”)
4. Mesa de trabalho multiusos.

Desenvolvimento do código

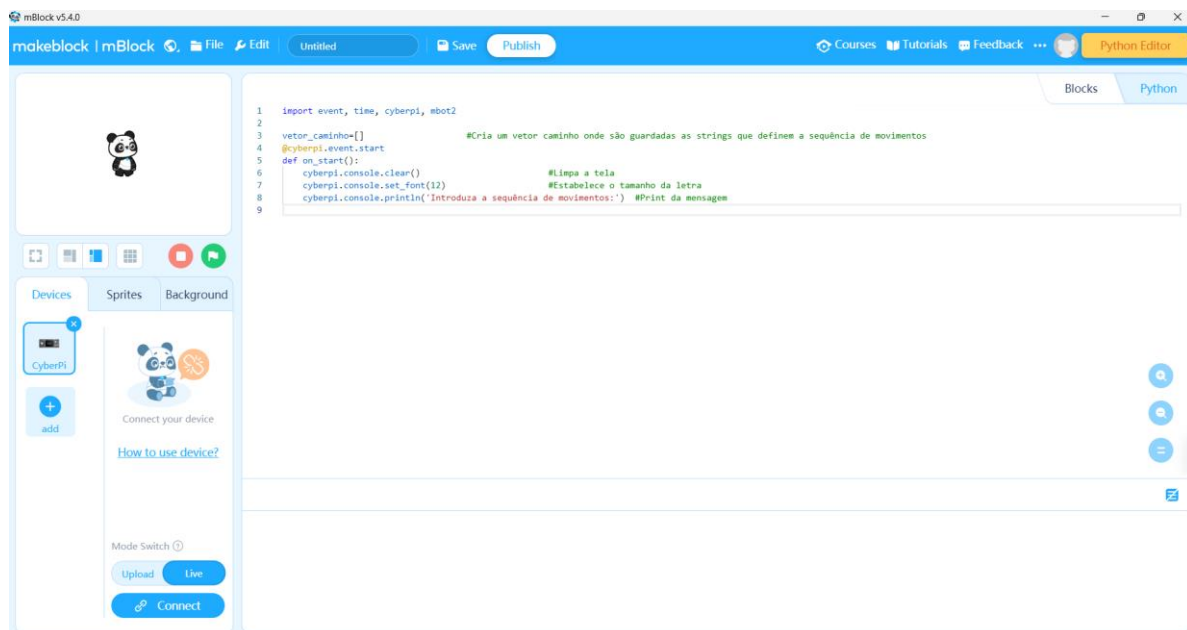
1. Abrir o mBlock no computador, abrir o editor Python.



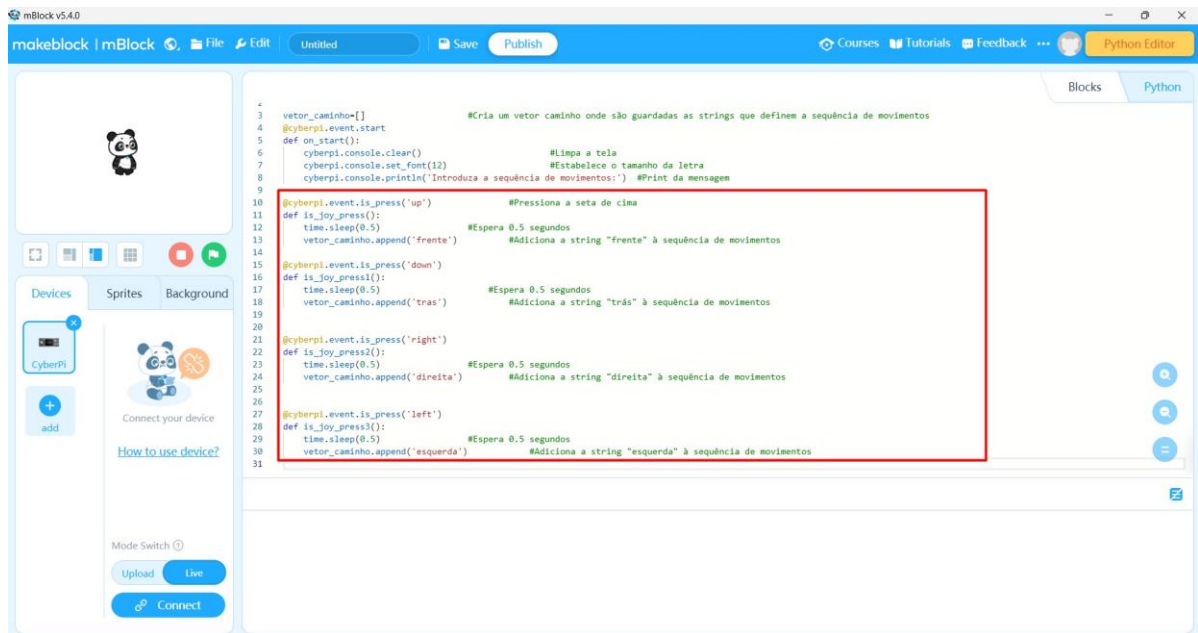
2. Importar as bibliotecas event, time, cyberpi e mbot2 que contém os comandos necessários para a realização do código.



3. Criar um vetor vazio (vetor_caminho) que irá guardar toda a sequência de movimentos. Estabelecer o que é impresso no ecrã quando o mBot2 é iniciado. No exemplo dado a mensagem é: 'Introduza a sequência de movimentos:'



4. Criar uma função para cada botão, dentro da função guardar no vetor_caminho uma string como movimento correspondente.



```
3 vetor_caminho=[] #Cria um vetor caminho onde são guardadas as strings que definem a sequência de movimentos
4 @cyberpi.event.start
5 def on_start():
6     cyberpi.console.clear() #Limpa a tela
7     cyberpi.console.set_font(12) #Estabelece o tamanho da letra
8     cyberpi.console.println('Introduza a sequência de movimentos:') #Print da mensagem
9
10 @cyberpi.event.is_press('up') #Pressiona a seta de cima
11 def is_joy_press1():
12     time.sleep(0.5) #Espera 0.5 segundos
13     vetor_caminho.append('frente') #Adiciona a string "frente" à sequência de movimentos
14
15 @cyberpi.event.is_press('down')
16 def is_joy_press1():
17     time.sleep(0.5) #Espera 0.5 segundos
18     vetor_caminho.append('tras') #Adiciona a string "tras" à sequência de movimentos
19
20
21 @cyberpi.event.is_press('right')
22 def is_joy_press2():
23     time.sleep(0.5) #Espera 0.5 segundos
24     vetor_caminho.append('direita') #Adiciona a string "direita" à sequência de movimentos
25
26
27 @cyberpi.event.is_press('left')
28 def is_joy_press3():
29     time.sleep(0.5) #Espera 0.5 segundos
30     vetor_caminho.append('esquerda') #Adiciona a string "esquerda" à sequência de movimentos
31
```

5. Criar uma função que é invocada quando o botão "A" é pressionado, dentro da função. Mostrar no ecrã qual a sequência de movimentos a realizar. Realizar a leitura do vetor e estabelecer uma correspondência entre a *string* e o respetivo movimento. No exemplo dado os movimentos para a frente e para trás são de 100 mm, esta distância deve ser ajustada em conformidade com o cenário construído.



```
22 @cyberpi.event.is_press('right')
23 def is_joy_press2():
24     time.sleep(0.5) #Espera 0.5 segundos
25     vetor_caminho.append('direita') #Adiciona a string "direita" à sequência de movimentos
26
27
28 @cyberpi.event.is_press('left')
29 def is_joy_press3():
30     time.sleep(0.5) #Espera 0.5 segundos
31     vetor_caminho.append('esquerda') #Adiciona a string "esquerda" à sequência de movimentos
32
33 @cyberpi.event.is_press('a')
34 def is_a_pressed():
35     cyberpi.console.println('Sequencia a realizar:')
36     for i in vetor_caminho:
37         cyberpi.console.println(i) #Mostra na tela a sequência de movimentos
38
39     for n in vetor_caminho:
40         #Ciclo que percorra o vetor caminho
41         if n=="frente":
42             #Verifica se a string do ciclo atual é "frente"
43             #Em caso afirmativo da condição anterior, realiza um movimento para a frente de 10 cm
44             mbot2.straight(10)
45             time.sleep(0.5) #Espera 0.5 segundos
46         if n=="tras":
47             #Verifica se a string do ciclo atual é "tras"
48             #Em caso afirmativo da condição anterior, realiza um movimento para trás de 10 cm
49             mbot2.straight(-10)
50             time.sleep(0.5) #Espera 0.5 segundos
51         if n=="direita":
52             #Verifica se a string do ciclo atual é "direita"
53             #Em caso afirmativo da condição anterior, realiza uma rotação sobre si próprio para a direita de 90 graus
54             mbot2.turn(90)
55             time.sleep(0.5) #Espera 0.5 segundos
56         if n=="esquerda":
57             #Verifica se a string do ciclo atual é "esquerda"
58             #Em caso afirmativo da condição anterior, realiza uma rotação sobre si próprio para a esquerda de 90 graus
59             mbot2.turn(-90)
60             time.sleep(0.5) #Espera 0.5 segundos
```

6. Testar o código e explicar aos alunos o objetivo, e o método para o atingir, instruindo a sequência de movimentos ao mBot.



Departamento de
Engenharia Eletromecânica

Protocolo Laboratorial

Robótica, Sistemas Robotizados, Robótica Industrial

Futebol

Departamento de Engenharia Eletromecânica
Universidade da Beira Interior

Covilhã e UBI, junho de 2023

Objetivos do protocolo

Este protocolo destina-se a guiar a montagem de uma atividade interativa, lúdica e cativante que não requer competências de programação. Adequado para visitas rápidas de alunos de todas as idades ao laboratório de robótica, o Futebot consiste num jogo simples e rápido onde os mBots são controlados através da aplicação Makeblock no telemóvel, agindo como jogadores num minijogo de futebol.

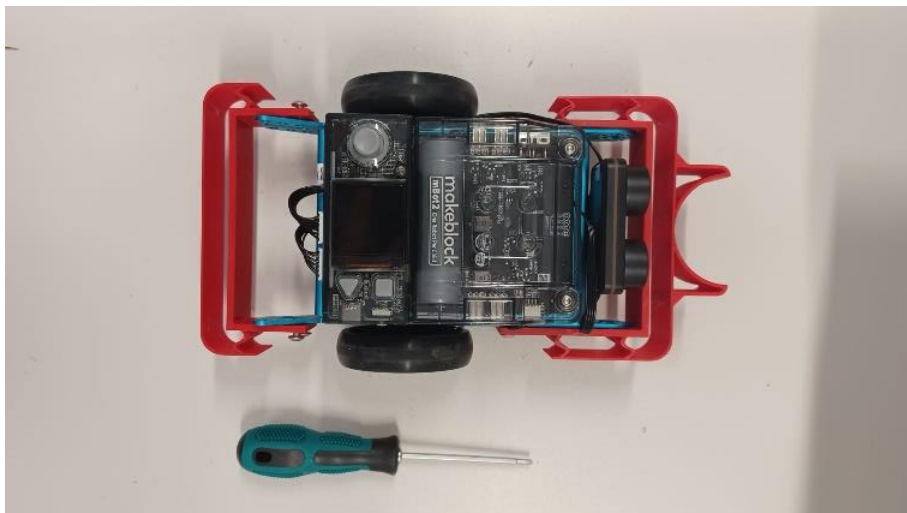
O documento explica a configuração da atividade, a instalação e o uso da aplicação Makeblock, além das regras do jogo. O objetivo é proporcionar uma experiência divertida e envolvente que permite aos visitantes interagir diretamente com os robôs, despertando assim o interesse pela robótica através de uma atividade competitiva.

Recursos necessários

1. Kit mBot2, incluindo cabos de conexão.
2. Carregador Type-C (5V).
Nota: A bateria do CyberPi é alimentada enquanto conectado ao computador.
3. Telemóvel com o mBlock 5 instalado (este passo será explicado neste documento).
4. Para-choques, balizas e bola (material que se encontra no laboratório).
5. Mesa de trabalho multiusos.

Preparação da atividade

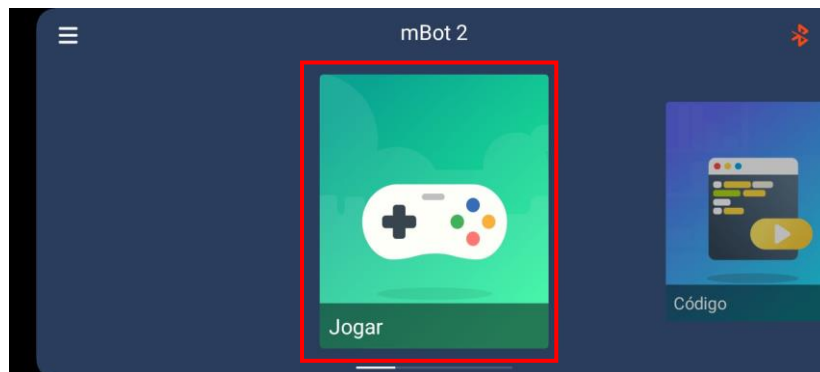
1. Instalar a aplicação Makeblock no telemóvel. pesquisar Makeblock App no Google Play (para android) ou App store (para ios) para fazer o download da aplicação.
2. Acoplar os para-choques nos mBot2, aparafusando-os nos furos presentes no chassi do mBot2, de acordo com a imagem abaixo:



3. Encaixar as balizas nos encaixes das laterais da mesa de trabalho multiusos para que a proteção do lado esquerdo e direito se encontrem equidistantes, isto é, as balizas têm de estar centradas.
4. Através da aplicação, conectar os mBot2 aos telemóveis dos participantes através de Bluetooth. Nota: Em dispositivos android é necessário ter a localização ativa neste passo. Posteriormente pode ser desligada esta opção para economizar bateria.
5. Selecionar o dispositivo desejado (mBot2):



6. Selecionar a opção “Jogar”:



7. Selecionar a opção “Dirigir”:



8. Utilizar o comando para dirigir o mBot2, utilizando os controles que se podem observar na imagem abaixo:



Recomenda-se que se utilizem velocidades mais lentas para dirigir o mBot com mais facilidade.

Regras do Futebot

1. No início de cada partida, os mBots devem estar encostados aos postes das respetivas balizas. A bola encontra-se no centro da linha de meio campo
2. As partidas terminam quando uma das equipas marcar 2 golos na baliza adversária ou a duração da partida atingir os 5 minutos. Em caso de empate, cumpre-se a regra do golo de ouro, onde a primeira equipa a marcar no tempo de compensação vence o jogo.
3. No retomar da partida, após uma das equipas marcar golo, todos os elementos do campo (mBots e bola) voltam à posição de início.
4. A partida é iniciada ou retomada iniciando uma contagem decrescente e os participantes apenas podem movimentar os robôs após o término da mesma.



Departamento de
Engenharia Eletromecânica

Protocolo Laboratorial

Robótica, Sistemas Robotizados, Robótica Industrial

Segue-linha

Departamento de Engenharia Eletromecânica
Universidade da Beira Interior

Covilhã e UBI, junho de 2023

Objetivos do protocolo

Este documento serve de apoio à realização da atividade segue-linha. Trata-se de um protocolo de introdução ao mBot2 à semelhança do protocolo “Configuração inicial e primeiros passos”, assim sendo todos os passos são descritos detalhadamente para melhor compreensão do aluno.

Propõe-se a realização de um trajeto com auxílio de um sensor seguidor de linha. Recomenda-se a utilização da pista e do sensor Quad RGB incluídos no kit do mBot2.

De seguida apresenta-se uma lista de passos para a preparação e realização da atividade.

Recursos Necessários

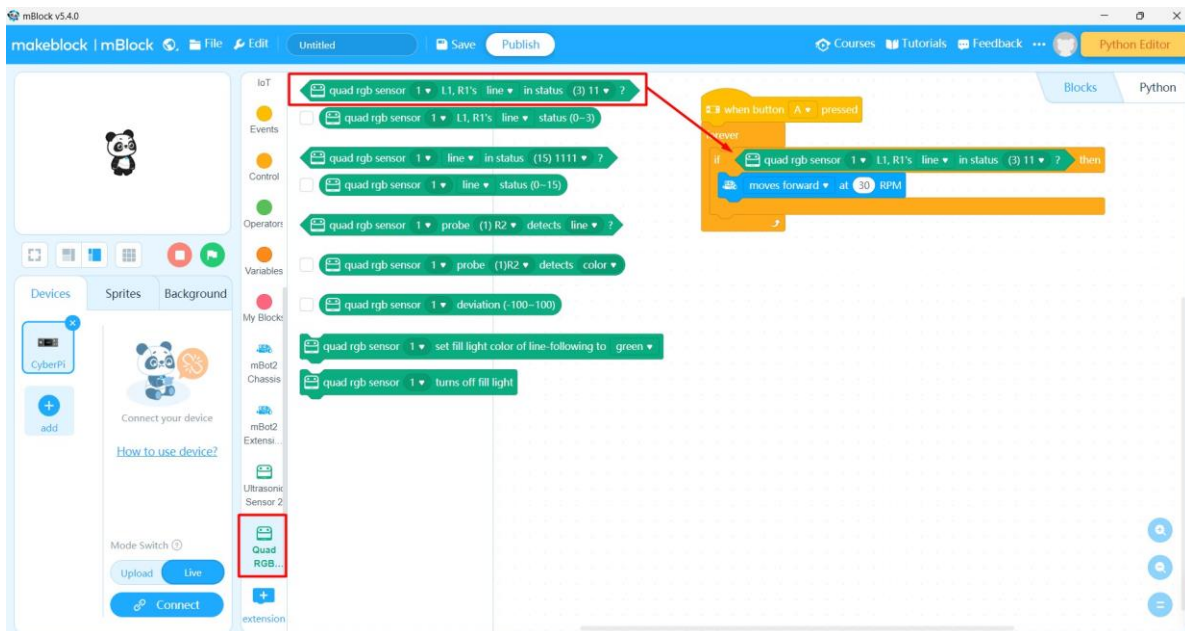
1. Kit mBot2, incluindo cabos de conexão, sensor de ultrassom e sensor Quad RGB
2. Carregador Type-C (5V).
Nota: A bateria do CyberPi é alimentada enquanto conectado ao computador.
3. Computador com mBlock 5 instalado (Ver Protocolo “Configuração inicial e primeiros passos”)
4. Mesa de trabalho multiusos.

Preparação da atividade

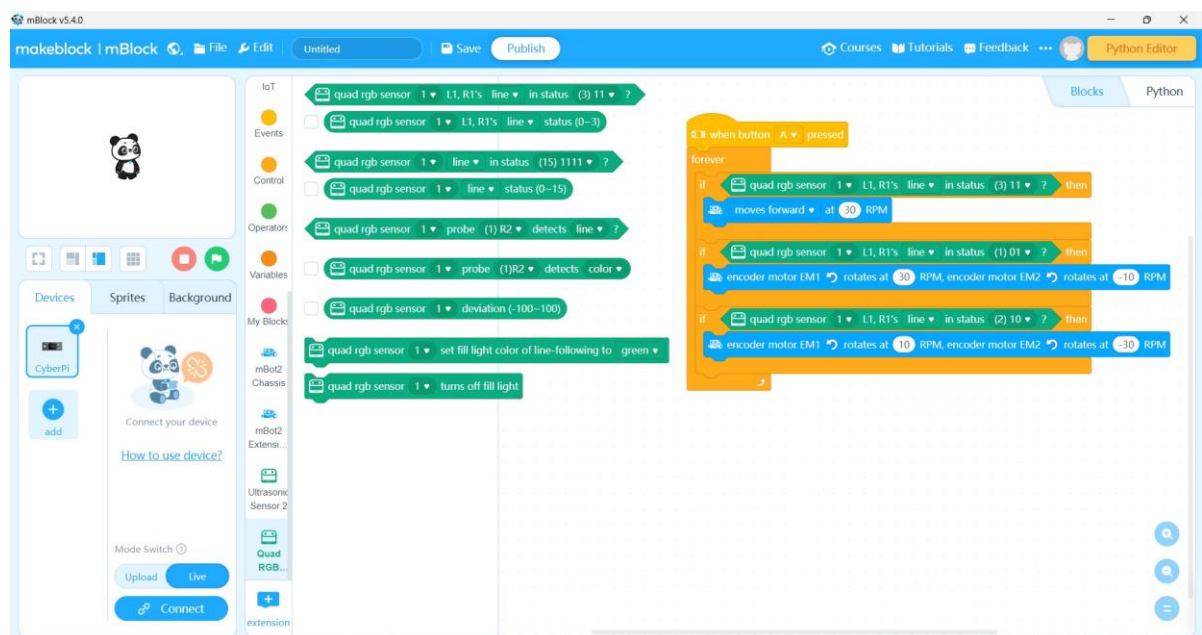
1. Retirar a pista proveniente do kit mBot2 e colocá-la na mesa de trabalho dedicada a atividades com kits de robótica móvel. Importante garantir que a pista fique plana, sem diferenças de nível que perturbem o movimento do robô.
2. Verificar se o sensor Quad RGB se encontra montado na parte inferior do chassi do mBot2 e as ligações do mesmo. Caso o sensor não se encontre instalado no robô deve seguir o manual de instruções presente no kit. Na página 11 encontra-se o processo de montagem e a disposição da cablagem.
3. Verificar se os parafusos que fixam as rodas e os respetivos motores exercem a força necessária para evitar oscilações durante a locomoção do robô. Na página 10 do manual de instruções encontra-se o processo de montagem.

Desenvolvimento do código

1. Abrir o mBlock5 no computador, adicionar todas as extensões necessárias para a manipulação do mBot2. Este passo é explicado detalhadamente no Protocolo “Configuração inicial e primeiros passos”
2. Recorrendo aos blocos contidos no conjunto "Control" criar a primeira condição. O sensor seguidor de linha contém quatro sensores RGB incorporados num único. Em todas as condições verifica-se o valor devolvido pelas variáveis L1 e R1 do sensor Quad RGB. Caso o valor de ambos seja 1, isto é, os dois sensores do meio encontram-se sobre a linha, é dada a instrução para o robô se mover para a frente.



3. As seguintes condições correspondem à correção da trajetória do mBot2. Caso o valor de L1 e R1 seja, respectivamente, 0 e 1, significa que o robô se encontra desalinhado. Existem várias formas de realizar o ajuste, uma delas será colocar a roda do lado esquerdo com uma velocidade de rotação superior à roda do lado direito. Nota para o sentido de rotação dos motores, pois eles encontram-se montados na mesma direção, mas em sentidos opostos. EM1 corresponde ao motor do lado esquerdo, EM2 corresponde ao motor do lado direito.



4. Ajustar e otimizar o código para que o mBot siga a linha de forma mais suave e veloz.



Departamento de
Engenharia Eletromecânica

Protocolo Laboratorial

Robótica, Sistemas Robotizados, Robótica Industrial

Desvio de obstáculos

Departamento de Engenharia Eletromecânica
Universidade da Beira Interior

Covilhã e UBI, junho de 2023

Objetivos do protocolo

Propõe-se aos alunos que coloquem o mBot Ultimate 2.0 a percorrer um trajeto. Ao longo deste trajeto será colocado um obstáculo de forma aleatória, o robô terá de ultrapassar essa adversidade, quer seja através da movimentação do objeto ou contornando-o.

Uma vez que existem diversas formas de realizar esta atividade, cada grupo deve planejar e construir um algoritmo próprio, apresentando um fluxograma do mesmo. O cenário fica igualmente à escolha dos alunos desde que possa ser integrado na mesa de trabalho multiusos.

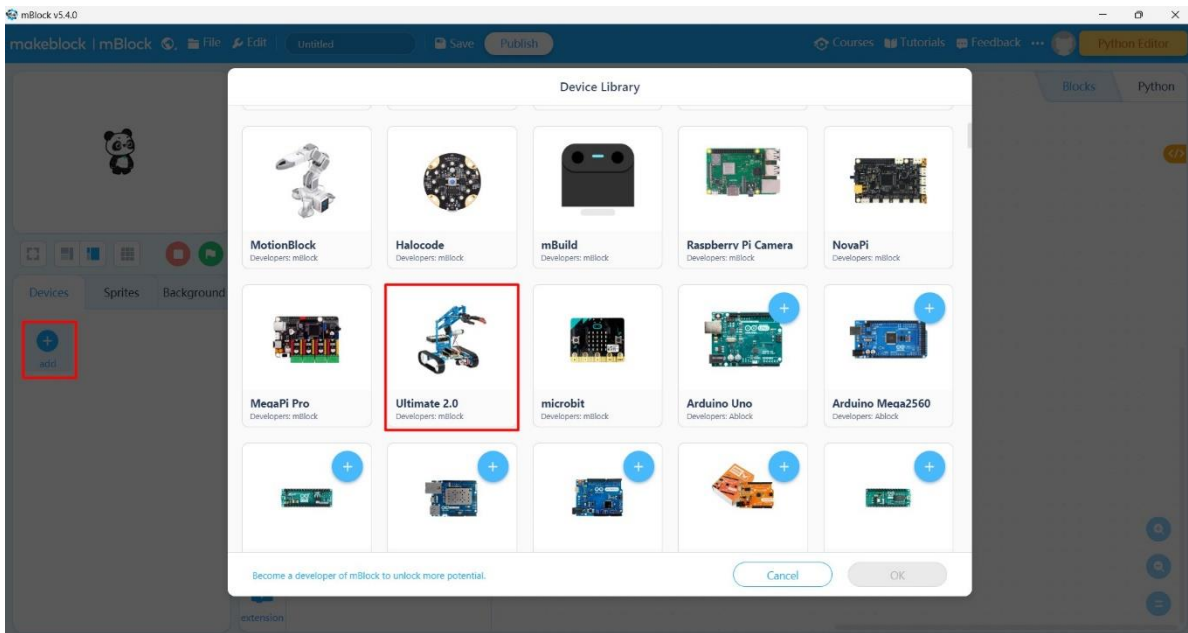
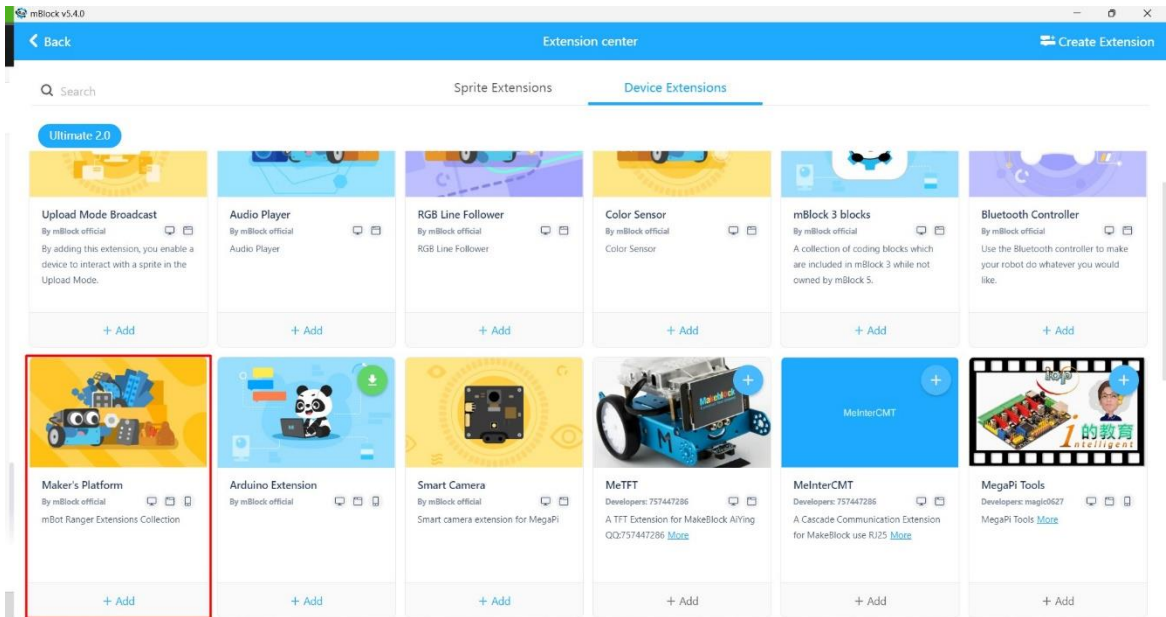
Neste documento são fornecidas algumas indicações para a realização da atividade.

Recursos necessários

1. Kit Ultimate 2.0, incluindo cabos de conexão, sensor de ultrassom e sensor seguidor de linha
2. Pilhas AA
3. Computador com mBlock 5 instalado (Ver Protocolo “Configuração inicial e primeiros passos”)
4. Fita de Papel (preferencialmente preta).
5. Mesa de trabalho multiusos.

Preparação da atividade

1. Na mesa de trabalho dedicada à robótica móvel, criar um trajeto com auxílio de fita de papel preta. A forma do trajeto fica ao critério dos alunos. Realçar a importância de utilizar fita de papel para que não permaneçam vestígios de cola ou tinta sobre a mesa pois todas as irregularidades interferem na leitura dos sensores presentes no robô.
2. Colocar as barreiras que o Ultimate 2.0 terá de ultrapassar.
3. Realizar as alterações necessárias na configuração "Robotic Arm Tank", isto é, os alunos podem implementar sensores seguidores de linha, sensores de ultrassom, entre outros. Para além disso, é permitido realizar alterações estruturais.
4. Abrir o mBlock no computador, adicionar o Ultimate 2.0 nos "Devices" e a extensão "Makers Platform", tal como indicado nas imagens abaixo.
5. Programar o robô para seguir a linha e evitar/desviar o obstáculo, continuando posteriormente a seguir a linha. Otimizar o código para que o faça no intervalo mais curto de tempo, e com o máximo de precisão possível. Em caso de dúvida, estudar novamente os protocolos “Configuração inicial e primeiros passos” e “Segue-linha”





Departamento de
Engenharia Eletromecânica

Protocolo Laboratorial

Robótica, Sistemas Robotizados, Robótica Industrial

Transporte de objetos

Departamento de Engenharia Eletromecânica
Universidade da Beira Interior

Covilhã e UBI, junho de 2023

Objetivos do protocolo

Propõe-se aos alunos que coloquem o mBot2 a percorrer um trajeto, durante o trajeto estará um objeto de pequenas dimensões (ex: uma esfera metálica). O robô terá de agarrar o objeto num ponto de recolha e movê-lo até ao ponto de entrega.

Uma vez que o mBot2 não possui no kit nenhuma ferramenta de manipulação, para esta atividade será fornecida uma garra desenvolvida com recurso à impressão 3D que lhe configura funções de manipulação com base nas saídas do mBot .

Existem diversas formas de programar esta atividade, em função do trajeto definido, cada grupo deve planear e construir um algoritmo próprio, apresentando um fluxograma do mesmo.

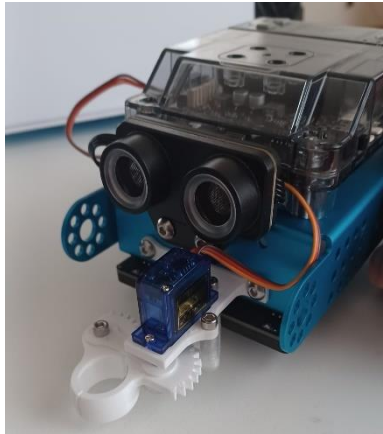
Neste documento são fornecidas algumas sugestões para a realização da atividade.

Recursos necessários

1. Kit mBot2, incluindo cabos de conexão, sensor de ultrassom e sensor Quad RGB
2. Garra desenvolvida com recurso à impressão 3D (escolher um kit que contenha uma garra)
3. Carregador Type-C (5V).
Nota: A bateria do CyberPi é alimentada enquanto conectado ao computador.
4. Computador com mBlock 5 instalado (na lista de passos está explícito como fazê-lo).
5. Fita de Papel (preferencialmente preta).
6. Mesa de trabalho multiusos.

Preparação da atividade

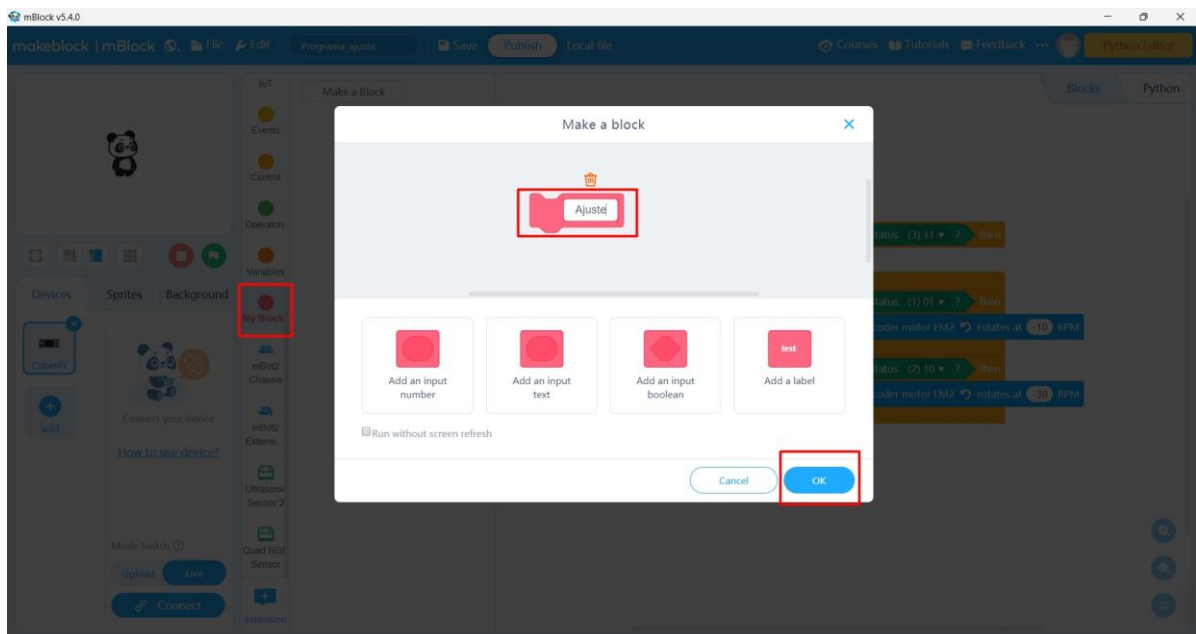
1. Na mesa de trabalho dedicada à robótica móvel, criar um trajeto com auxílio de fita de papel preta. A forma do trajeto fica ao critério dos alunos. Realçar a importância de utilizar fita de papel para que não permaneçam vestígios de cola ou tinta sobre a mesa pois todas as irregularidades interferem na leitura dos sensores presentes no robô.
2. No trajeto definido destacar dois locais, o primeiro local será o ponto de recolha do objeto e o segundo local será o ponto de entrega. Para distinguir os locais pode ser utilizada fita de papel com outras cores (ex: vermelho, verde, azul).
3. Verificar se o sensor Quad RGB está corretamente instalado e se o conjunto rodas+motores estão bem fixos de forma a evitar oscilações durante a locomoção do robô. (Ver Protocolo “Segue-linha”)
4. Montar a garra impressa fornecida, de acordo com a figura seguinte:



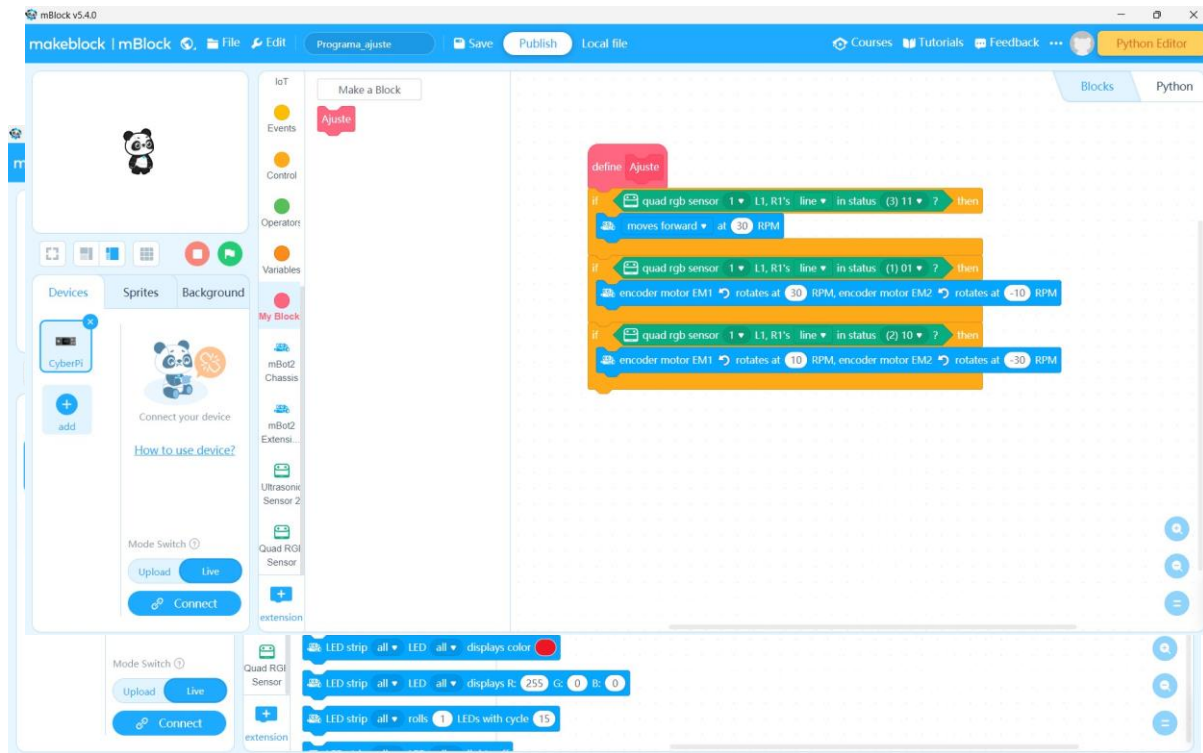
Desenvolvimento do código

1. Criar uma função para realizar o ajuste da trajetória do robô (Caso seja necessário ver Protocolo “Segue-linha”).

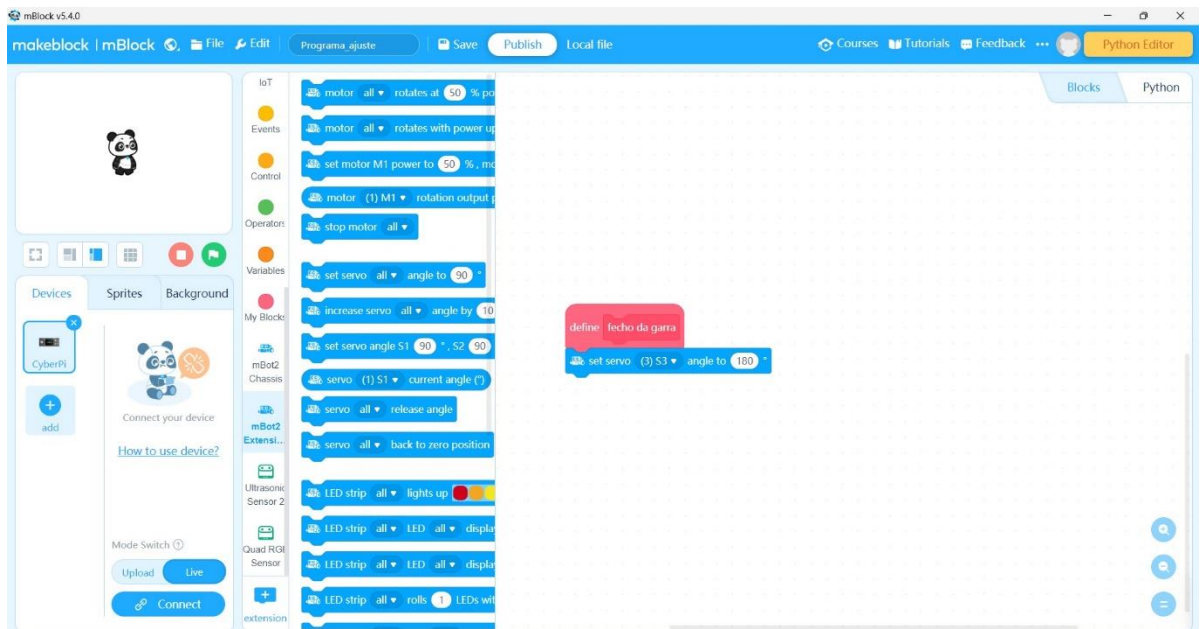
Para criar uma função, podemos aceder a "My Blocks" e atribuir um nome ao novo bloco criado.



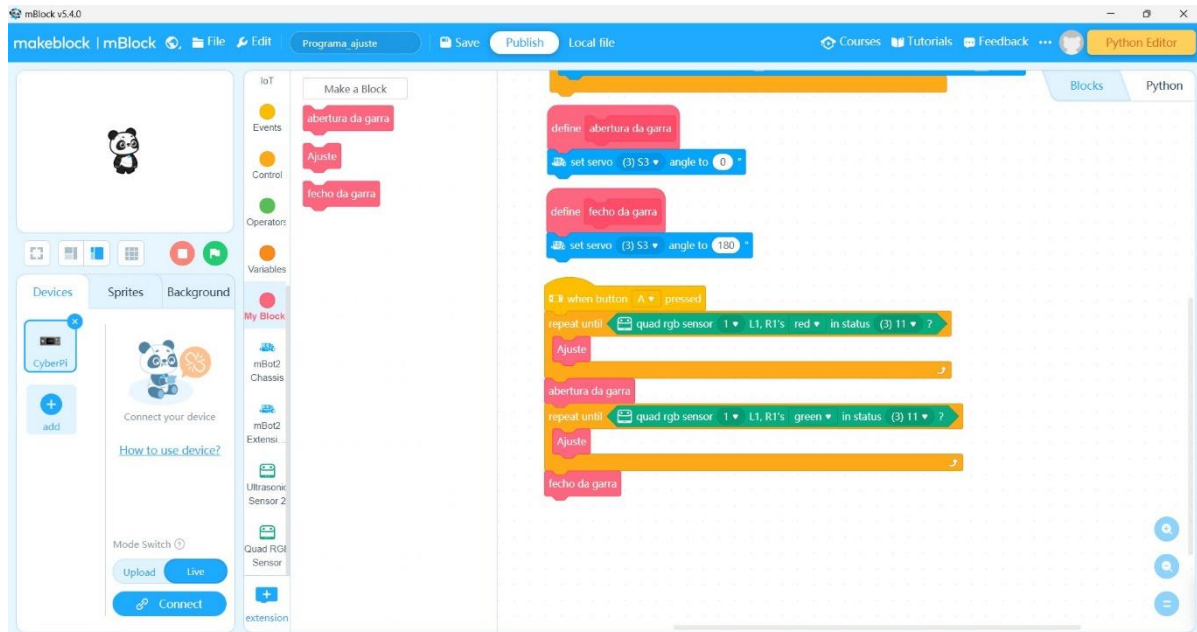
Colocar as instruções pretendidas por baixo do bloco que define a função.



2. Repetir o passo anterior, mas desta vez criar funções para o controlo da garra. A abertura e o fecho da garra é feito através dos blocos que estão no conjunto "mBot2 Extension".



3. Invocar as funções no código principal em função da cor que é detetada pelo sensor Quad RGB, suponhamos que a tira de fita vermelha se encontra no local de recolha e a fita verde no local de entrega.



Ajustar os parâmetros e otimizar o código para aumentar a velocidade e precisão de manipulação da esfera.



Departamento de
Engenharia Eletromecânica

Protocolo Laboratorial

Robótica, Sistemas Robotizados, Robótica Industrial

Torre de blocos

Departamento de Engenharia Eletromecânica
Universidade da Beira Interior

Covilhã e UBI, junho de 2023

Objetivos do protocolo

O objetivo deste protocolo é utilizar o Ultimate 2.0, com a configuração "Robotic Arm Tank", a construir uma torre com pelo menos 3 blocos de altura.

O percurso e o método da construção fica ao critério do grupo desde que o objetivo proposto seja alcançado.

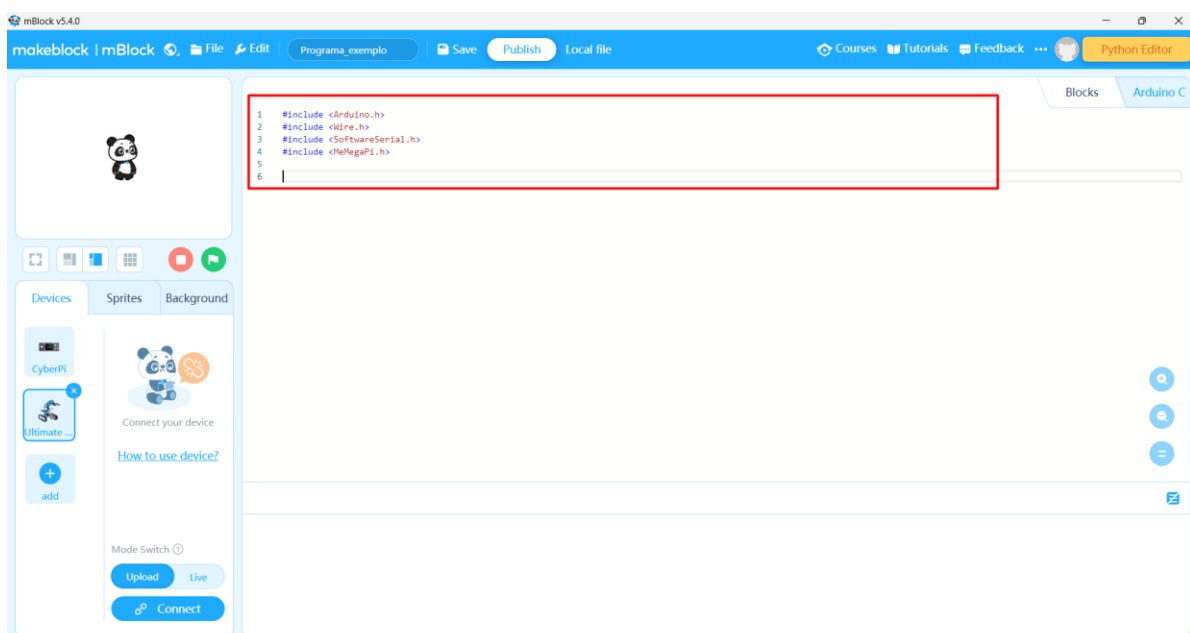
O script para a realização desta atividade será apresentado em C++ (Arduino)

Recursos necessários

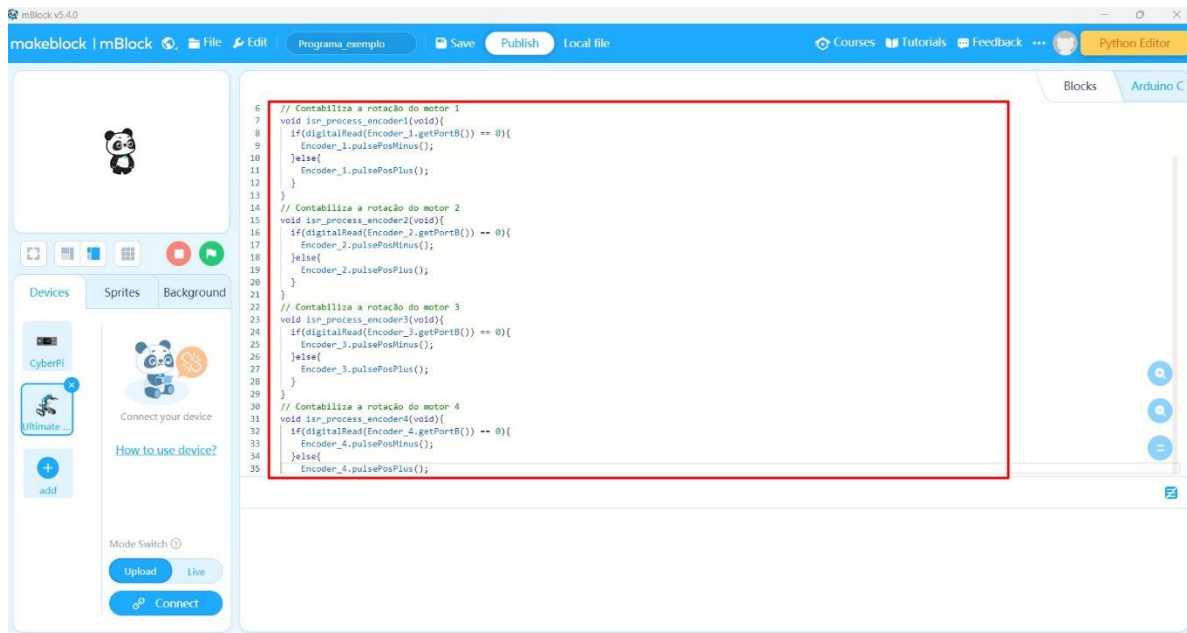
1. Kit Ultimate 2.0, incluindo cabos de conexão, sensor de ultrassom e sensor seguidor de linha
2. Pilhas AA
3. Computador com mBlock 5 instalado (Ver Protocolo "Configuração inicial e primeiros passos")
4. Fita de Papel (preferencialmente preta).
5. Blocos de madeira.
6. Mesa de trabalho multiusos.

Sugestão de realização da atividade

1. Para programar o Ultimate 2.0 em C++, é necessário utilizar o editor Arduino disponibilizado pela aplicação mBlock ou instalar o Arduino IDE e adicionar as bibliotecas da Makeblock. O segundo método encontra-se explicado detalhadamente em: <https://www.yuque.com/makeblock-help-center-zh/ultimate-senior/arduino-programming/>.
2. Recorrendo ao editor Arduino do mBlock, o primeiro passo é adicionar as bibliotecas que contém os comandos essenciais para a manipulação dos motores do robô (Arduino.h, Wire.h, SoftwareSerial.h, MeMegaPi.h)

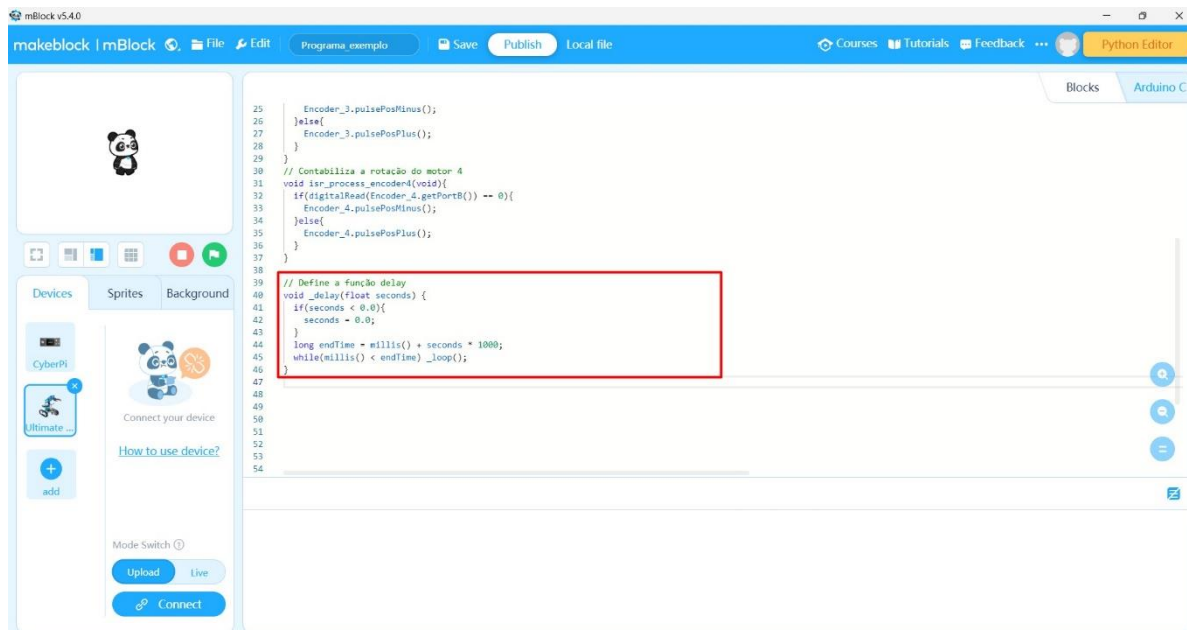


3. Definir as funções que contabilizam a rotação de cada motor.



```
6 // Contabiliza a rotação do motor 1
7 void isr_process_encoder1(void){
8   if(digitalRead(Encoder_1.getPortB()) == 0){
9     Encoder_1.pulsePosMinus();
10  }else{
11    Encoder_1.pulsePosPlus();
12  }
13 }
14 // Contabiliza a rotação do motor 2
15 void isr_process_encoder2(void){
16   if(digitalRead(Encoder_2.getPortB()) == 0){
17     Encoder_2.pulsePosMinus();
18   }else{
19     Encoder_2.pulsePosPlus();
20   }
21 }
22 // Contabiliza a rotação do motor 3
23 void isr_process_encoder3(void){
24   if(digitalRead(Encoder_3.getPortB()) == 0){
25     Encoder_3.pulsePosMinus();
26   }else{
27     Encoder_3.pulsePosPlus();
28   }
29 }
30 // Contabiliza a rotação do motor 4
31 void isr_process_encoder4(void){
32   if(digitalRead(Encoder_4.getPortB()) == 0){
33     Encoder_4.pulsePosMinus();
34   }else{
35     Encoder_4.pulsePosPlus();
36   }
37 }
```

4. Definir a função delay



```
25 Encoder_3.pulsePosMinus();
26 }else{
27   Encoder_3.pulsePosPlus();
28 }
29 }
30 // Contabiliza a rotação do motor 4
31 void isr_process_encoder4(void){
32   if(digitalRead(Encoder_4.getPortB()) == 0){
33     Encoder_4.pulsePosMinus();
34   }else{
35     Encoder_4.pulsePosPlus();
36   }
37 }
38
39 // Define a função delay
40 void _delay(float seconds) {
41   if(seconds < 0.0){
42     seconds = 0.0;
43   }
44   long endTime = millis() + seconds * 1000;
45   while(millis() < endTime) _loop();
46 }
47
48
49
50
51
52
53
54
```

5. Configurar o comando dos motores, tal como indicado na imagem que se segue:

```

28 }
29 }
30 // Contabiliza a rotação do motor 4
31 void isr_process_encoder4(void){
32   if(digitalRead(Encoder_4.getPortB()) == 0){
33     Encoder_4.pulsePosPlus();
34   }else{
35     Encoder_4.pulsePosPlus();
36   }
37 }
38
39 // Define a função delay
40 void _delay(float seconds) {
41   if(seconds < 0.0){
42     seconds = 0.0;
43   }
44   long endTime = millis() + seconds * 1000;
45   while(millis() < endTime) _loop();
46 }
47
48 // Configura o PWM dos motores
49 void setup() {
50   attachInterrupt(Encoder_1.getIntNum(), isr_process_encoder1, RISING);
51   attachInterrupt(Encoder_2.getIntNum(), isr_process_encoder2, RISING);
52   TCCR1A = _BV(WGM10); // Esta linha configura o registro do controle do Timer 1. O bit WGM10 está é ativado para definir o modo de operação do Timer 1 como "Fast PWM 8-bit".
53   TCCR1B = _BV(CS11) | _BV(WGM12); // Aqui, o registro do controle do Timer 1 (TCCR1B) é configurado. Os bits CS11 e WGM12 são ativados. O CS11 define a configuração do prescaler
54   TCCR2A = _BV(WGM21) | _BV(WGM20); // Esta linha configura o registro do controle do Timer 2 (TCCR2A). Os bits WGM21 e WGM20 são ativados para definir o modo de operação do Tim
55   TCCR2B = _BV(CS21); // Aqui, o registro do controle do Timer 2 (TCCR2B) é configurado. O bit CS21 define a configuração do prescaler como 8, o que define a frequência de conta
56 }

```

6. Definir a estratégia a utilizar. No exemplo que se segue não está a resolução da atividade, contudo ilustra os comandos de manipulação dos motores em função das leituras do sensor de ultrassons.

```

50   attachInterrupt(Encoder_1.getIntNum(), isr_process_encoder1, RISING);
51   attachInterrupt(Encoder_2.getIntNum(), isr_process_encoder2, RISING);
52   TCCR1A = _BV(WGM10); // Esta linha configura o registro do controle do Timer 1. O bit WGM10 está é ativado para definir o modo de operação do Timer 1 como "Fast PWM 8-bit".
53   TCCR1B = _BV(CS11) | _BV(WGM12); // Aqui, o registro do controle do Timer 1 (TCCR1B) é configurado. Os bits CS11 e WGM12 são ativados. O CS11 define a configuração do prescaler
54   TCCR2A = _BV(WGM21) | _BV(WGM20); // Esta linha configura o registro do controle do Timer 2 (TCCR2A). Os bits WGM21 e WGM20 são ativados para definir o modo de operação do Tim
55   TCCR2B = _BV(CS21); // Aqui, o registro do controle do Timer 2 (TCCR2B) é configurado. O bit CS21 define a configuração do prescaler como 8, o que define a frequência de conta
56   while(!(ultrasonic_8.distanceCm() < 10)) // O robô movimenta-se para a frente enquanto a distância medida pelo sensor ultrassom for superior a 10 cm
57   {
58     _loop();
59
60     Encoder_1.setTarPWM(50/100.0*255); // Atribui uma potência de 50% ao motor 1
61     Encoder_2.setTarPWM(50/100.0*255); // Atribui uma potência de 50% ao motor 2
62     if(ultrasonic_8.distanceCm() > 20){ // O robô movimenta o braço se a distância medida pelo sensor ultrassom for superior a 20 cm
63
64       Encoder_3.setTarPWM(50/100.0*255); // Atribui uma potência de 50% ao motor 3
65       _delay(1);
66
67       Encoder_3.setTarPWM(-50/100.0*255); // Atribui uma potência de -50% ao motor 3 (inverte o sentido de rotação)
68       _delay(1);
69
70       Encoder_3.setTarPWM(0/100.0*255); // Cessa o movimento do motor 3
71     }
72   }
73   Encoder_1.setTarPWM(0); // Cessa o movimento do motor 1
74   Encoder_2.setTarPWM(0); // Cessa o movimento do motor 2
75 }
76
77
78 void _loop() {
79   Encoder_1._loop();
80   Encoder_2._loop();
81   Encoder_3._loop();
82 }
83
84 void loop() {
85   _loop();
86 }
87 }

```

No programa apresentado, o robô movimenta-se para a frente enquanto a distância medida pelo sensor de ultrassons for superior a 100 mm. Dentro do ciclo é verificada uma condição: Se a distância medida for superior a 200 mm, o braço do robô realiza um movimento ascendente, seguido de um movimento descendente.

7. Adaptar o programa fornecido para detetar e manipular os. Isto implica uma calibração de distâncias que vão variar de acordo com o *setup* inicial, e definição de uma rotina de manipulação que permite empilhar verticalmente pelo menos três blocos.



Departamento de
Engenharia Eletromecânica

Protocolo Laboratorial

Robótica, Sistemas Robotizados, Robótica Industrial

Labirinto

Departamento de Engenharia Eletromecânica
Universidade da Beira Interior

Covilhã e UBI, junho de 2023

Objetivos do protocolo

O objetivo deste protocolo é aplicar um algoritmo que coloque o mBot2 a percorrer qualquer configuração possível do labirinto. O labirinto na mesa de trabalho multiusos será montado com as placas MDF (5 mm de espessura) e respectivos suportes que se encontram no laboratório. A configuração do labirinto pode ser alterada a qualquer altura para testar a resiliência do algoritmo a diferentes configurações.

A estratégia utilizada fica ao critério de cada grupo, contudo é exigido que o código seja escrito em Python.

Recursos necessários

1. Kit mBot2, incluindo cabos de conexão.
2. Carregador Type-C (5V).
Nota: A bateria do CyberPi é alimentada enquanto conectado ao computador.
3. Computador com mBlock 5 instalado (Ver Protocolo “Configuração inicial e primeiros passos”).
4. Fita de papel (3 cores diferentes).
5. Placas MDF (5mm) e suportes de encaixe das mesmas.
6. Mesa de trabalho multiusos.

Sugestão de realização da atividade

1. Na mesa de trabalho multiusos, com auxílio de fita de papel (preferência cor preta para facilitar a leitura do sensor Quad RGB), fazer uma grelha cujas linhas passem no centro, entre duas paredes adjacentes do labirinto, como ilustra a figura seguinte:



2. Sinalizar o início e o fim do labirinto com fita de papel utilizando uma cor diferente da grelha (ex: vermelho, azul, verde).

3. Consultar a biblioteca de comandos para o editor Python disponibilizada pela Makeblock em: <https://education.makeblock.com/help/category/mblock-python/mblock-python-editor-python-api-documentation-for-devices/mblock-python-editor-python-api-documentation-for-cyber-pi/>
4. Programar o robô para seguir a linha utilizando o sensor Quad RGB e, detectar as paredes com o sensor de ultrassons.
5. Com base na informação medida, aplicar um algoritmo que permita a solução do labirinto, de forma mais eficiente e no menor espaço de tempo possível.
6. Para aumentar a eficiência da detecção de paredes, podem ser acrescentados mais sensores de ultrassons, ou desenvolvido um mecanismo com recurso aos outputs do mBot e um servomotor, que permita uma maior velocidade de detecção.



Departamento de
Engenharia Eletromecânica

Protocolo Laboratorial

Robótica, Sistemas Robotizados, Robótica Industrial

Condução autónoma

Departamento de Engenharia Eletromecânica
Universidade da Beira Interior

Covilhã e UBI, junho de 2023

Objetivos do protocolo

Este protocolo visa ser integrado com um dos projetos habitualmente realizados na unidade curricular de Automação Industrial, o projeto de um sistema de controlo de semáforos, com interrupção gerada pelo peão através de um controlador lógico programável (PLC).

Os mBot2 simulam o papel de veículos numa situação de tráfego normal, para isso é necessário dotar o robô com a capacidade de tomar decisões em função das luzes emitidas pelos semáforos. Esta possibilidade é adicionada com incorporação do módulo *Smart Camera*, que permite adicionar visão computacional às capacidades do mBot2.

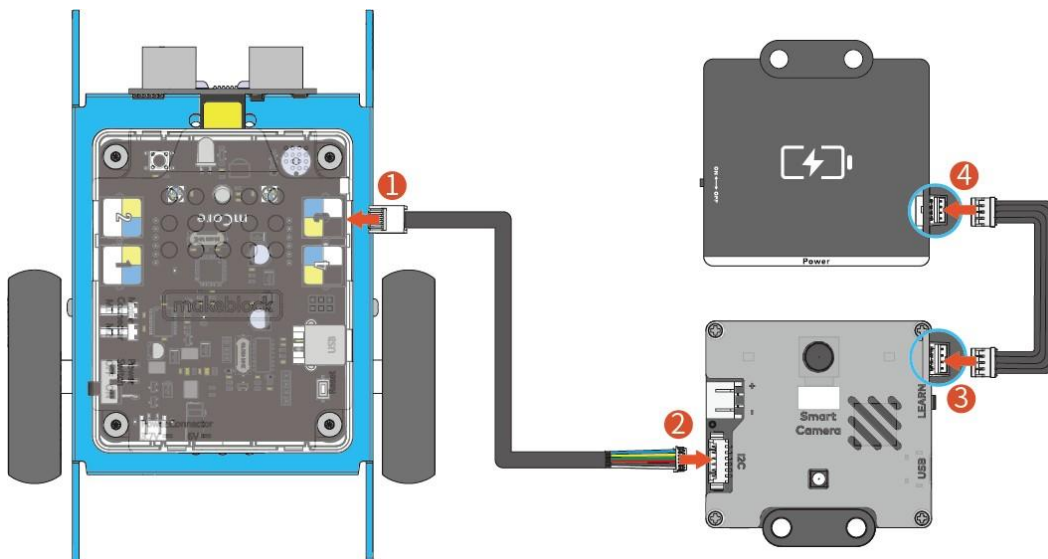
O script do código deverá ser realizado em Python e resultar num movimento aleatório do mBot2 pela pista definida previamente, de forma aleatória, e obedecendo aos sinais luminosos.

Recursos necessários

1. Kit mBot2, incluindo cabos de conexão, sensor de ultrassom e sensor Quad RGB
2. *Smart Camera* e respetivos cabos de conexão (Na preparação da atividade existe um esquema de conexão)
3. Carregador Type-C (5V).
Nota: A bateria do CyberPi é alimentada enquanto conectado ao computador.
4. Computador com mBlock 5 instalado (na lista de passos está explícito como fazê-lo)
5. Mesa de trabalho multiusos.
6. Semáforos previamente montados no âmbito da disciplina de Automação Industrial.

Preparação da atividade

1. Preparar o cenário de acordo com o projeto dos semáforos na mesa de trabalho multiusos
2. Instalar a Smart Câmera no mBot2 de acordo com o manual de instruções. Se necessário, projetar uma estrutura para fixar a câmera e recorrer à impressão 3D para a produzir.



3. No desenvolvimento do script, importar as bibliotecas event, time, cyberpi, mbot2 e mbuild.
4. Consultar a biblioteca da Makeblock com os comandos a utilizar no editor Python, disponíveis em: <https://education.makeblock.com/help/category/mblock-python/mblock-python-editor-python-api-documentation-for-devices/mblock-python-editor-python-api-documentation-for-cyberpi/>
5. Programar o mBot para se deslocar aleatoriamente pelo percurso, obedecendo aos sinais luminosos: avançando perante o sinal verde, reduzindo a velocidade perante o sinal amarelo e travando perante o sinal vermelho, até que volte a surgir o sinal verde.