

ENNARD – Robô de Rescue Maze da Escola de Robótica e Mecatrônica ABC Marcelo Salles

Felipe Catapano Emrich Melo, André Luiz Buzacarini Schulman, Marcelo Ribeiro Salles

Abstract – This article presents the development of an Arduino-controlled autonomous junior robot with emphasis on Latin American Robotics Competition 2018. All key components and critic design choices, as well as their purposes are detailed in this document.

I. INTRODUÇÃO

Pelo fato de estarmos iniciando na categoria e por complicações nas circunstâncias e envolvendo o projeto, nosso robô competidor para a LARC de 2017 (edição ocorrida em Curitiba, PR) fora construído com diversos problemas estruturais e soluções improvisadas. Utilizávamos motores de plástico energizados por um pack de pilhas (rompido internamente no meio da competição e consequentemente anulando uma de nossas rodadas), que eram fracos e tinham sua movimentação controlada por tempo (o que gerava um alto grau de imprecisão variável). Além disso a disposição dos sensores formava grandes pontos cegos.

Havendo, no entanto, superadas muitas das dificuldades e havendo observado inúmeros outros projetos de âmbito internacional em várias competições, o novo agente foi desenvolvido com a solução por completo da missão proposta em mente, que envolve a simulação de um salvamento de vítimas quentes e visuais em um ambiente de desastre inacessível a seres humanos.



Imagem 1 - LESION, o robô de 2017

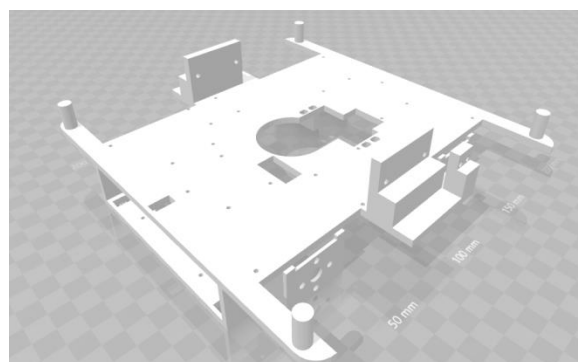


Imagem 2 - Chassi

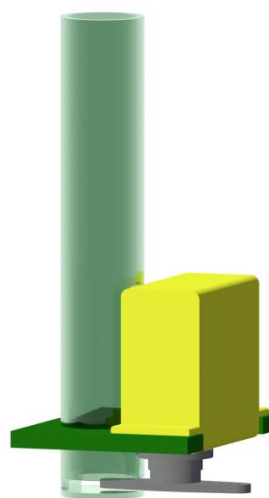


Imagem 3 - Depositador

Para similar finalidade, a de reduzir ao máximo a quantidade de fios e evitar problemas facilmente, percebemos que a impressão de circuitos integrados era essencial (também a desenhamos com AutoCAD). Duas PCBs ligadas por grupos de flat cables permitem que todas as trilhas possuam 1cm de largura e sejam consideravelmente expostas, reduzindo a possibilidade de erros.

II. ESTRUTURA E CIRCUITOS

Com a utilização de uma impressora 3D, somos capazes de projetar e criar estruturas com encaixes perfeitos para os componentes de forma rápida e barata, portanto o chassi do robô foi desenhado com o software AutoCAD e impresso em PLA amarelo. O tubo e kits do nosso depositador de resgate, no entanto, foram feitos em acrílico para que não houvesse qualquer rebarba.

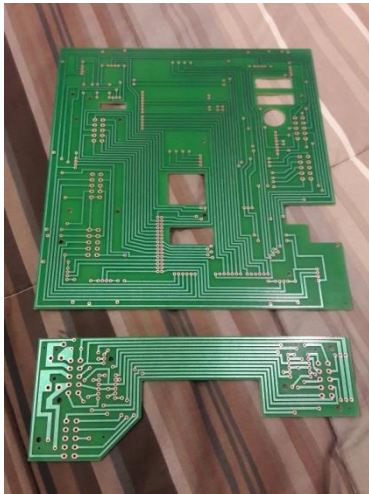


Imagem 4 – Circuitos integrados (primeiros protótipos)

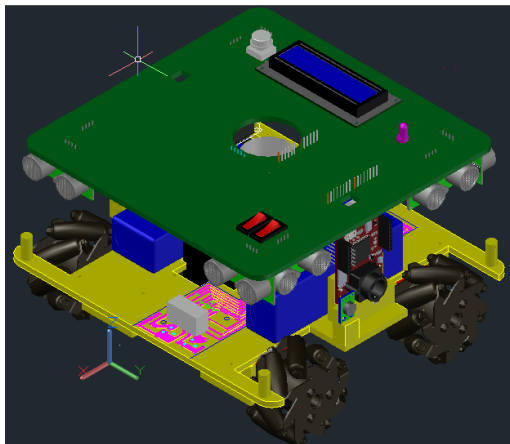


Imagem 5 – Robô completo simulado em CAD

III. CONTROLE E ATUAÇÃO

Arduino DUE foi a plataforma central escolhida pela simplicidade de uso, maior capacidade de variáveis e maior velocidade em relação a todos os outros arduinos, além da possibilidade de utilizar qualquer pino digital com *interrupts*, para melhor aproveitamento dos encoders presentes nos motores por exemplo (é a única placa da Arduino com arquitetura ARM). Ela (juntamente com os sensores conectados) é alimentada por um pack de baterias de Li-Ion, enquanto os motores são controlados por duas pontes-h L298n alimentadas em série por uma bateria de Li-Po separada.

Grande parte de nossa estratégia se dá pela escolha dos motores DC de 12V da Pololu, colocando a qualidade de alinhamento sensorial e velocidade de ultrapassagem de obstáculos acima da precisão absoluta garantida por servos motores. Um servo motor genérico MG996R (visto no suporte acima) foi utilizado para o depósito dos kits de resgate pelo baixo peso e custo para seu tamanho.

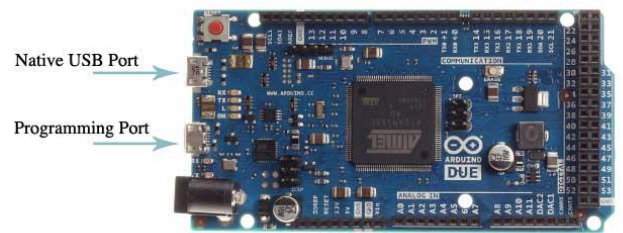


Imagem 6 – Arduino DUE



Imagem 7 – Motor com encoder

E para que fosse possível garantir total liberdade de movimento ao robô, as rodas ligadas aos motores são rodas mecanum da Nexus, de 6cm de diâmetro, adaptadas por calços usinados customizados para este fim. Seus rolos dispostos em 45 graus permitem a mesma característica de um sistema com quatro rodas omni tradicionais: Facilmente mover-se em qualquer direção (útil para alinhamentos rápidos e movimentos pequenos), porém com o benefício de poderem ser dispostas de maneira reta como rodas comuns.



Imagem 8 – Roda mecanum

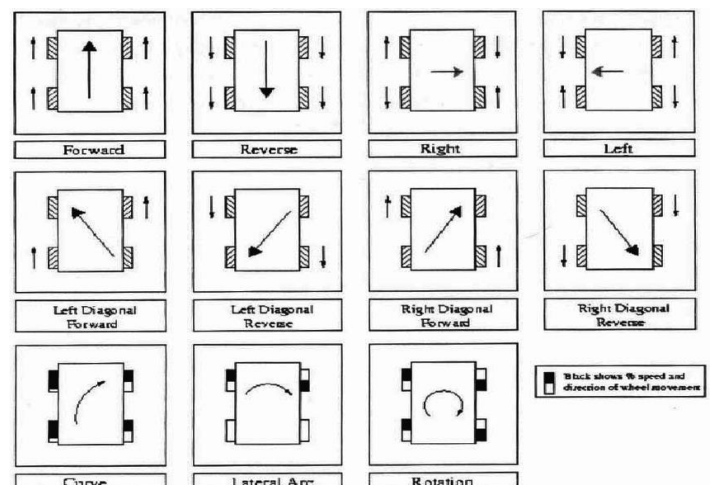


Imagem 9 – Esquema de movimentação com 4 rodas mecanum

IV. SENSORES E SINALIZAÇÃO

Para possibilitar a identificação das novas vítimas visuais, utilizamos duas pequenas placas programáveis OpenMV M7, ligadas ao arduino por comunicação UART. Por possuírem conexão externa e uma IDE própria, calibrar as placas pode ser feito de maneira direta e é possível retirá-las do robô facilmente caso necessário.



Imagem 10 – OpenMV com cartão SD

Escolhemos usar nove sensores ultrassônicos US-015 dispostos na horizontal para o sensoriamento das paredes pelo seu amplo campo de visão e precisão de medida acima dos comuns HC-SR04. Por suportarem o uso de *interrupts* digitais e não precisarmos ler todos as instâncias de uma única vez, seu elevado delay em relação a um sensor de luz torna-se desprezível.

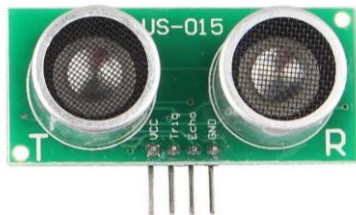


Imagem 11 – Sensor ultrassônico

A medição inercial é realizada por um único módulo 10DOF, o GY-87, que reúne giroscópio, acelerômetro(MPU6050, utilizado para percepção da rampa) e magnetômetro(HMC5883L, utilizado como bússola), além de um barômetro.

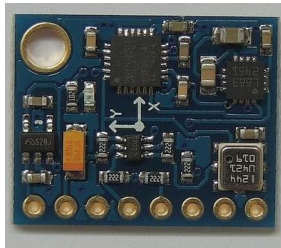


Imagem 12 – Unidade de Medição Inercial (UMI)

Para completar o sensoriamento utilizamos os populares MLX90614 para o reconhecimento das vítimas quentes, e dois sensores TCRT5000 para refletância do solo (perceber tiles pretos e tiles de checkpoint), além dos encoders presentes nos motores.

Há uma tela LCD de 16x2 divisões para feedback por texto, o que dispensa a constante necessidade de manter uma conexão com o computador para rastrear alguma variável ou

processo, além de um LED multicolorido para sinalizar ações ao juiz e ao capitão da equipe com apenas um único sinal.

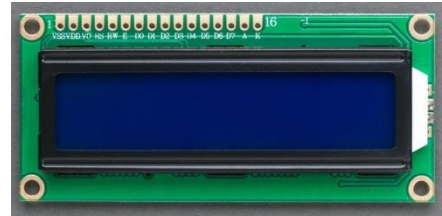


Imagem 13 – Display com luz de fundo azul

V. O SOFTWARE

Toda a programação do arduino foi escrita com a linguagem C++ utilizando o editor de texto Sublime Text 3 e compilada com a ArduinoIDE. Esta visa uma movimentação rápida e totalmente contínua do robô (isto é, sem um único delay) com alinhamentos constantes em relação às paredes. Um algoritmo de controle Proporcional Integral Derivativo (PID) será implementado para manter o robô centralizado ao subir e descer as rampas.

Um filtro Low Pass é aplicado na maioria das leituras para que não haja muitos ruídos entre as mesmas, e .

Para o *path planning* (decisão de quais direções a tomar dentro do labirinto), utilizamos o algoritmo de Tremaux como base (virar à direita é priorizado em relação a seguir em frente ou virar à esquerda, com objetivo de “contornar” o mapa), e a complementamos com um algoritmo de Floodfill modificado (priorização de áreas ainda não exploradas na rodada e retorno ao quadrado inicial pelo caminho mais curto conhecido, sem a necessidade de rotinas secundárias ou um mapeamento da configuração do labirinto). Na partida, Um mapa virtual é criado com todos os nodos inicialmente zerados, e, após o robô passar de um nodo a outro, caso o valor deste for zero ele passa a ser o sucessor do valor do nodo anterior.

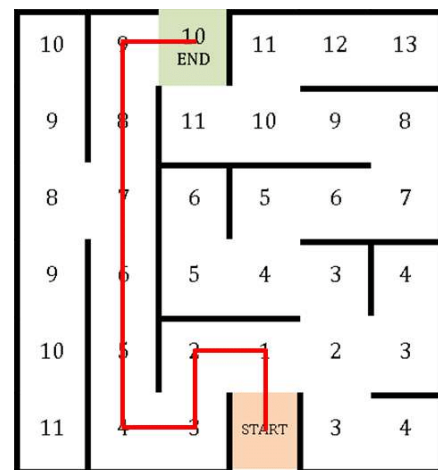


Imagem 14 – Exemplo de FloodFill resolvendo um labirinto com saída

A programação das câmeras é escrita em Python (elas funcionam com o sistema operacional MicroPython) pela própria IDE da OpenMV, e consiste na comparação entre cada frame capturado pelo módulo embutido e pequenas fotos de letras presentes na memória flash (algoritmo conhecido como *Template Matching*). Parâmetros básicos de imagem como exposição, equilíbrio de branco, etc.são automaticamente

normalizados pela câmera, assim reduzindo todos os parâmetros humanamente ajustados a apenas três filtros de 0 a 1, um para cada letra a ser identificada.

VI. ARENA DE TESTES

Percebemos durante o planejamento deste projeto que a construção de um ambiente que simule situações oficialmente presente nas competições é essencial para uma boa preparação da estratégia. Portanto a equipe investiu em uma arena que seguisse fielmente as regras, apesar do seu real tamanho quando montada, para podermos testar aspectos do robô antes mesmo de este estar funcionando por completo.

A madeira é o material ideal utilizado na grande maioria desses eventos pelo seu baixo custo, resistência e facilidade de produzir encaixes lineares. Portanto toda a estrutura da arena de testes é feita com este material. Lombadas, garrafas de areia, tijolos, varetas e outros obstáculos estão a disposição para criar cenários com um nível de desafio equivalente ou maior ao visto na RoboCup Junior.

VII. CONCLUSÃO

Foi possível perceber que a parte mais importante de um projeto deste porte é a fase de pesquisa, que nos permitiu estabelecer contato com diversas opções de dispositivos de melhor performance para a realização desta e de outras tarefas da engenharia mecatrônica e da ciência da computação, descartar estratégias/metodologias que demonstraram menos resultados antes mesmo do início da montagem do robô, comunicar-se com outros competidores e no geral evoluir como estudantes juniores.

VIII. AGRADECIMENTOS

A equipe ENNARD Salles gostaria muito de agradecer à Escola de Robótica, a seus professores e envolvidos, e aos nossos pais por nos propiciar esta incrível oportunidade e ferramentas para trabalhar com o Resgate e aos ex-competidores Leonardo Santander da Silva e Ivan Seidel Gomes pelas experiências compartilhadas e por serem de grande fonte de inspiração.

IX. REFERÊNCIAS

- [1] Escola de Robótica e Mecatrônica ABC Marcelo Salles. Disponível em: <http://www.roboticaabc.com.br>
- [2] RoboCup Junior. Disponível em: <http://junior.robocup.org>
- [3] Arduino platform. Disponível em: <https://www.arduino.cc>
- [4] OpenMV platform. Disponível em: <https://openmv.io>
- [5] MicroPython. Disponível em: <https://micropython.org>
- [6] NEXUS Robot. Disponível em: <http://www.nexusrobot.com>
- [7] Impressão 3D. Disponível em: https://pt.wikipedia.org/wiki/Impressão_3D
- [8] AutoCAD. Disponível em: <https://www.autodesk.com.br/products/autocad/overview>
- [9] Sublime Text. Disponível em: <https://www.sublimetext.com>
- [10] Ponte H. Disponível em: https://pt.wikipedia.org/wiki/Ponte_H
- [11] Motores Pololu de 25Dmm. Disponível em: <https://www.pololu.com/category/115/25d-mm-metal-gearmotors>
- [12] Servo Motor. Disponível em: <https://pt.wikipedia.org/wiki/Servomotor>
- [13] US-015 vs HC-SR04. Disponível em: <https://www.youtube.com/watch?v=aLkkAsrSibo&t=219s>
- [14] Display de Cristal Líquido (LCD) . Disponível em: <https://pt.wikipedia.org/wiki/LCD>
- [15] IMU. Disponível em: https://en.wikipedia.org/wiki/Inertial_measurement_unit
- [16] Ultrassônico. Disponível em: <https://pt.wikipedia.org/wiki/Ultrassom>
- [17] Temperatura Infravermelha. Disponível em: <http://guias.oxigenio.com/como-funciona-um-termometro-infravermelho>
- [18] Luz Infravermelha. Disponível em: https://pt.wikipedia.org/wiki/Radia%C3%A7%C3%A3o_infravermelha
- [19] Algoritmo Trémaux. Disponível em: https://pt.wikipedia.org/wiki/Algoritmo_de_Tremaux
- [20] Algoritmo Floodfill. Disponível em: https://en.wikipedia.org/wiki/Flood_fill
- [21] Material para arena de testes. Disponível em: <http://www.leomadeiras.com.br>