

Conversão de um carrinho de controle remoto, para ser controlado remotamente utilizando o Raspberry

Rafael Marcelo dos Santos¹, Daniela Flôr¹

¹Campus Paranavaí – Instituto Federal do Paraná (IFPR) Paranavaí - PR – Brasil

rafikrafael@gmail.com, daniela.flor@ifpr.edu.br

Abstract. *The objective of this article is to develop a small scale car prototype, converting a remote control car and making it capable of being guided remotely through a web page in a browser accessed by a computer or cell phone, sending directional commands using the keyboard arrows or the direction buttons available on the web page. Incorporating Raspberry to the prototype, making it capable of running the application of independent control of another device.*

Resumo. *O objetivo deste artigo é o desenvolvimento de um protótipo de carro em escala reduzida, convertendo um carrinho de controle remoto e tornando-o capaz de ser guiado remotamente através de uma página web, em um browser acessado por um computador ou celular, enviando comando direcionais utilizando as setas do teclado ou os botões de direção disponíveis na página web. Incorporando o Raspberry ao protótipo, tornando-o capaz de executar a aplicação de controle independente de outro dispositivo.*

1. Introdução

Nos últimos anos as pesquisas em veículos autônomos ou semiautônomos encontra-se em constante desenvolvimento. Pesquisas em áreas científicas e não científicas buscam soluções para os problemas no trânsito, visando torna-lo mais seguro, prático e eficiente.

Com o intuito de elaborar um projeto de um carro guiado remotamente em escala reduzida, tendo como foco conhecer semelhanças e comportamentos de veículos autônomos, o projeto propõe a conversão de um carrinho de controle remoto para um carrinho controlado por um Raspberry PI. Foram estudadas tecnologias e métodos para a construção e ou adaptação do hardware, do desenvolvimento do aplicativo que provê a comunicação entre o Raspberry PI e o Arduino, e que forneça a interface de controle para usuário, por meio de uma rede sem fio.

A realização desse projeto impulsiona o posterior desenvolvimento de protótipos mais complexos, através da implantação de novas funcionalidades como ser capaz de desviar de obstáculos, capacidade de estacionar e se locomover sozinho de forma autônoma.

2. Revisão da Literatura

O principal fundamento para a construção do protótipo, é o estudo sobre carros semiautônomos e autônomos. A possibilidade de substituir a condução humana de veículos atual para condução automatizada, parcial ou completa, de forma mais segura, realizada por sistemas computacionais, através de hardwares, sensores, ferramentas para o processamento de informações em tempo real, tomando decisões que tornem o trânsito mais seguro.

Os primeiros experimentos começaram a partir de 1950, em 1970 as pesquisas por melhores radares foram intensificadas, e já a partir de 1990 a comercialização de

radars automotivos foi inicializada, radars com tecnologia com ultrassom, câmeras de vídeo e outros. Nesse mesmo período já existiam nos Estados Unidos sistemas de detecção de colisão para veículos. Em 1999 a Mercedes Benz já introduzia em seu modelo “Classe S” o uso de radars do tipo ACC, que auxiliavam o condutor em situações em caso de distração no trânsito, alerta de possível colisão, assistentes para estacionamento alertando a distancia a objetos e outros. Em 2005 empresas como BMW e Volkswagen introduziram como item opcional em alguns modelos sensores baseados em radars ACC [Schneider, 2005].

Em 2003 a DARPA, Organização Central de Pesquisa e Desenvolvimento do Departamento de Defesa dos Estados Unidos, anunciou o primeiro desafio DARPA Grand Challenger na área de automação de veículos, o desafio constituía em desenvolver um veículo autônomo capaz de percorrer um percurso em um terreno off-road. Foi composto por participantes de instituições universitárias e centros de pesquisas particulares, a cada ano o número de participantes sempre aumentava [Seetharaman et al., 2006]. Tal evento refletiu em um aumento na busca e disseminação de conhecimento na área de automação veicular, tanto para fins comerciais quanto militares.

O projeto CADU (Carro Autônomo Desenvolvido na UFMG), de acordo com [Sabbagh, 2009], é um veículo autônomo baseado em um automóvel de passeio, está sendo desenvolvido desde 2007 pelo Grupo de Pesquisa e Desenvolvimento de Veículos Autônomos (PDVA) da Universidade Federal de Minas Gerais, e em 2009 já tinha em seus projeto a utilização de mecanismos de controle automático de frenagem, aceleração, câmbio e direção.

Devido o alto custo no desenvolvimento de um projeto utilizando um automóvel de passeio, são desenvolvidos protótipos em escala reduzida, utilizando carrinhos de controle remotos em varias escalas, ou desenvolvendo toda a estrutura do protótipo. Foram analisados alguns artigos de projetos com carrinhos de controle remoto, como o “RC to Rpi Car Conversion” do Departamento de Ciências da Computação da Northern Illinois University [Myers, 2017]. E o “DrivingMatter: An Autonomous RC Car using Raspberry Pi” do Departamento de Ciências da Computação da National University of Computer and Emerging Sciences [Chishti et al., 2018].

Desta forma, devido a relevância do tema em questão e a crescente discussão acerca de veículos autônomos, justifica-se o desenvolvimento desse artigo, baseado em um projeto de simulação de um automóvel de passeio em pequena escala, com custos e riscos de acidentes reduzidos em comparação ao de um veículo de passeio.

3. Metodologia

Nos tópicos abaixo serão expostas os hardwares utilizados e fases de desenvolvimento do projeto.

3.1. Hardware e estrutura mecânica

Como o proposito do artigo é a conversão de um carrinho de controle remoto e não a montagem completa da estrutura do protótipo, foi escolhido um carrinho de controle remoto elétrico pequeno, capaz de comportar os componentes necessários, que possui bateria recarregável com tensão de 7.5 volts, suas dimensões são de 30 cm de comprimento, 13 cm de altura e 13 cm de largura. Com uma estrutura plástica capaz de suportar as adaptações necessárias em sua estrutura, além de peso extra. A Figura 1 apresenta o carrinho utilizado.



Figura 1. Carrinho de Controle Remoto. Fonte: Autor.

O projeto pretende que o carrinho substitua o controle remoto por processamento computacional executado pelo próprio protótipo, sendo independente de hardware externo e capaz de disponibilizar o acesso através da aplicação de controle remoto pela rede sem fio. Por esse motivo foi utilizado o Raspberry PI 3 model B.

Raspberry PI é um minicomputador com arquitetura multitarefa, que é a possibilidade de uso de mais de uma aplicação ao mesmo tempo. Tem suporte a diversas linguagens de programação, como Java, Python, NodeJs, etc, ferramentas como Git Subversion para efetuar a gerencia da versão dos códigos fontes desenvolvidos. É possível efetuar o desenvolvimento da aplicação diretamente no Raspberry PI, utilizando um teclado e mouse conectados à entrada USB, e monitor à saída HDMI. De acordo com [Raspberry PI, 2018], as especificações técnicas do Raspberry PI model B, incluem um processador Quad Core 1.2GHz Broadcom BCM2837 64bit, memória RAM, rede 10/100 Ethernet e 2.4GHz 802.11n wireless, bluetooth, uma entrada para cartão microSD, com 4 portas USBs, 1 HDMI, e uma entrada micro USB de 5 volts a 2.5 amperes.

No Raspberry PI foram instalados e configurados os aplicativos necessários para o controle do carrinho de forma remota.

Para o controle do protótipo e as conexões de eventuais módulos, foi utilizado o Arduino Mega, devido a maior robustez de seus componentes em questões de falhas e problemas de tensão nos circuitos. Ele atua como uma ponte de comunicação entre o Raspberry PI e os hardwares. E sua comunicação com o Raspberry PI ocorre pela porta USB através de conexão serial, para execução de comandos de direção que o protótipo deverá seguir, e envio de informações ao Raspberry PI.

De acordo com [Arduino, 2018], as especificações técnicas do Arduino Mega incluem um micro controlador ATmega2560, 16 entradas analógicas, 54 pinos de entradas e saídas digitais, sendo 15 destas saídas do tipo PWM. [Hirzel, 2018] descreve PWM como uma técnica utilizada por sistemas digitais para variação do valor médio de uma forma de onda periódica quadrada, ela mantém a frequência da onda e controla o tempo em que ela fica em nível lógico alto, essa variação é chamada de Duty Cycle, no Arduino ela utiliza valores de 0 a 255.

Para o controle dos motores do protótipo foi utilizado um módulo de Ponte H, ligada ao Arduino Mega. Como descrito por [Reis, 2017], Ponte H é um circuito que permite realizar a inversão da polaridade da corrente que flui através de uma carga, com isso controlando a direção de rotação dos motores. Ela contém duas saídas que permite o controle de dois motores, duas entradas de tensão para sinal PWM que permitem controlar a velocidade da rotação, uma entrada GND, uma entrada de alimentação que

possui capacidade de 3 a 30 volts de tensão, e uma saída de 5 volts que pode ser usada para alimentar um Arduino ou outro componente.

A Figura 2 apresenta o módulo de Ponte H.

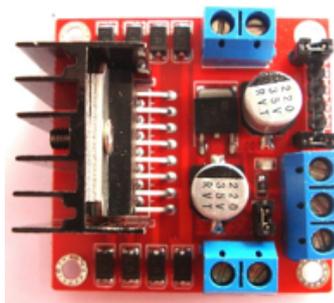


Figura 2. Módulo Ponte H. Fonte: Baú da Eletrônica

Como descrito por [Reis, 2017], para o funcionamento da Ponte H, são utilizadas quatro chaves eletrônicas (S1, S2, S3, S4), que podem ser controladas de forma independente. Dependendo da configuração de acionamento entre as chaves, a corrente percorre o motor ora por um sentido, ora por outro.

A Figura 3 representa as chaves da Ponte H.

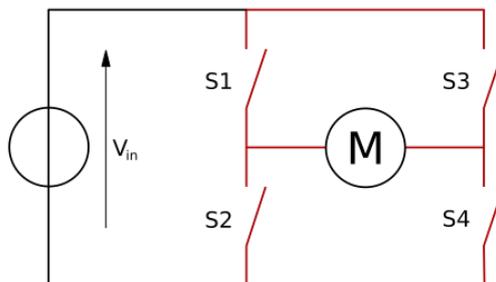


Figura 3. Diagrama das chaves da Ponte H. Fonte: Cyril Buttay.

O estado das chaves e suas combinações determinam o funcionamento do motor, as combinações podem ser feitas de várias formas, por exemplo: se nenhum par de chaves está acionado o motor está desligado, acionando as chaves S1 e S4 o motor gira para um sentido e ao acionar as chaves S2 e S3 o motor gira para o lado oposto. A Figura 4 representa as ligações das chaves [Reis, 2017].

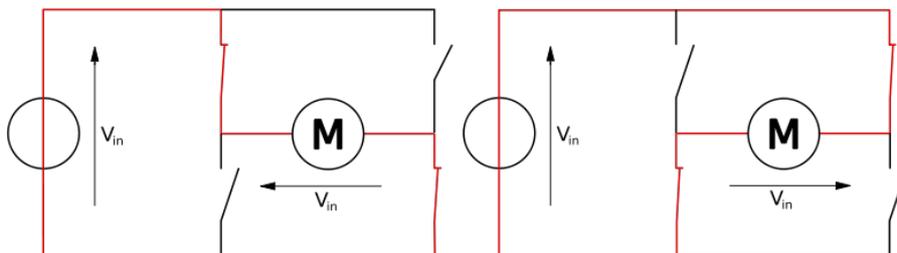


Figura 4. Exemplo das ligações das chaves da Ponte H. Fonte: Cyril Buttay.

Para demonstrar o funcionamento do controle de velocidade de rotação do motor, através do PWM no protótipo, foi alimentada a Ponte H com 7.5 volts, foi dado um comando através da entrada “A” da Ponte H para rodar o motor para um lado de forma constante. Desta forma os três casos abaixo demonstram a aplicação de três valores de pulso aplicado ao pino PWM da entrada “A”.

O primeiro pulso foi enviado com valor 0, neste caso na saída “A” da Ponte H aplica a tensão de 0 volts, tal aplicação resultou na inércia do motor. A Figura 5 apresenta o exemplo do pulso de 0 volts.



Figura 5. Exemplo de pulso PWM tensão 0 volts. Fonte: Timothy Hirzel.

O segundo pulso foi enviado com valor 127, metade do valor máximo suportado, neste caso na saída “A” da Ponte H aplica a tensão de 3.75 volts, tal aplicação resultou na rotação do motor com velocidade média. A Figura 6 apresenta o exemplo do pulso de 3.75 volts.

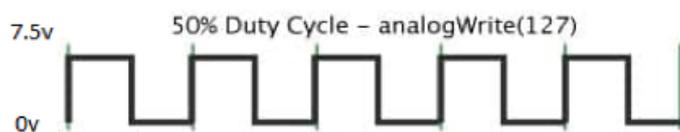


Figura 6. Exemplo de pulso PWM tensão 3.75 volts. Fonte: Timothy Hirzel.

O terceiro pulso foi enviado com valor 255, valor máximo suportado, neste caso na saída “A” da Ponte H aplica a tensão de 7.5 volts, tal aplicação resultou na rotação máxima do motor. A Figura 5 apresenta o exemplo do pulso de 7.5 volts.

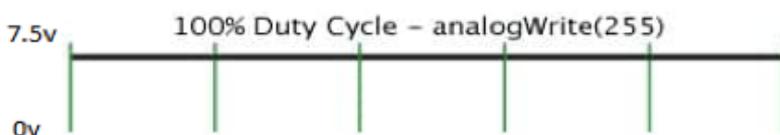


Figura 7. Exemplo de pulso PWM tensão 7.5 volts. Fonte: Timothy Hirzel.

O Raspberry PI necessita de uma fonte de alimentação confiável. A versão utilizada do Raspberry PI necessita que a tensão seja de 5 volts com 2.5 Amperes. A tensão é recomendada ter variação de 5%, sendo de 4.75 a 5.25 volts. Para evitar danificar a placa é imprescindível a utilização de uma fonte de energia confiável que forneça a tensão exigida com 2 amperes. Foi utilizado o carregador portátil Tl-pb10400 da marca TP-Link, as especificações técnicas são: capacidade de 10400Ah, entrada de 5 volts 2 amperes, duas saídas com máxima de 5 volts a 2.4 amperes e um total máximo de 5 volts a 3 [TP-Link, 2018].

3.2. Montagem do protótipo

Para a montagem do protótipo foi necessário efetuar a remoção da carcaça principal do carrinho, deixando-o somente no eixo, tornando possível o acesso aos seus componentes e circuitos. Para tornar o carrinho controlado pelo Raspberry PI, foi necessário retirar a placa controladora que vem de fábrica com o carrinho, permitindo a ligação da Ponte H para controle dos motores.

Para extrair a placa controladora original foi necessário remover os parafusos de fixação, as soldas dos fios que ligam a placa aos motores e aos fios de alimentação dos polos positivo e negativo da bateria. A Ponte H foi posicionada próximo ao motor frontal, fixando-a na carcaça com cola quente, foi colada as extremidades da placa ao chassi. Após a secagem da cola quente, foi ligada a fiação do motor frontal diretamente à saída “A” da Ponte H.

A próxima alteração foi a retirada da estrutura da transmissão traseira, permitindo a troca da fiação do motor traseiro, e do botão de ligar e desligar do carrinho que fica ligado ao polo positivo da bateria. Foram retiradas as soldas da fiação original e efetuadas novas, instalando a fiação com maior comprimento, capazes de alcançar a posição que foi instalada a Ponte H.

Finalizada a troca dos fios e reinstalada a transmissão traseira na estrutura do chassi, foi efetuada a conexão dos fios à Ponte H, a fiação do motor traseiro foi conectada à saída “B”. Na entrada GND recebeu o fio do polo negativo da bateria e um jumper, que é um fio do tipo “macho x fema”, para ser conectado ao GND do Arduino fechando o circuito. Na entrada VCC recebeu o fio do polo positivo da bateria do carrinho. A Figura 8 apresenta as ligações.

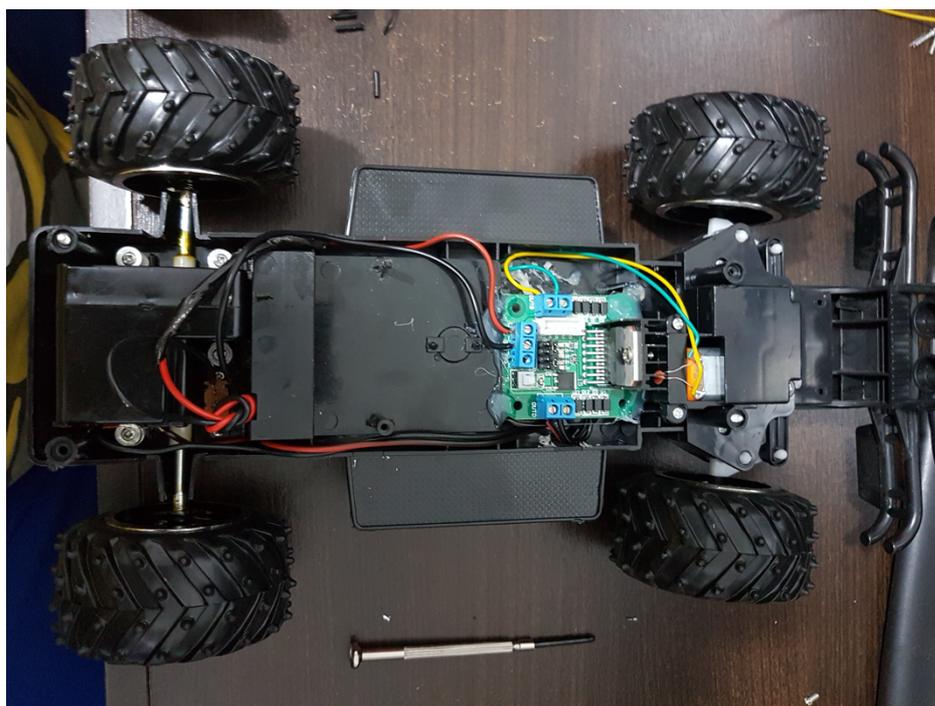


Figura 8. Instalação da Ponte H no chassi. Fonte: Autor.

Para acomodar o Arduino no chassi, foi necessário efetuar alguns cortes no suporte superior removível do chassi utilizando um estilete, deixando um espaço para os fios dos jumpers.

O Arduino foi colocado próximo a Ponte H, foi utilizado jumpers do tipo “macho x fema” para a ligação entre o Arduino e a ponte H, foi ligado o motor frontal nos pinos 47, 49 e o 3 no PWM, o motor traseiro foi ligado nos pinos 51, 53 e o 2 no PWM.

A Figura 9 apresenta o digrama com todas as ligações efetuadas entre o Arduino e Ponte H.

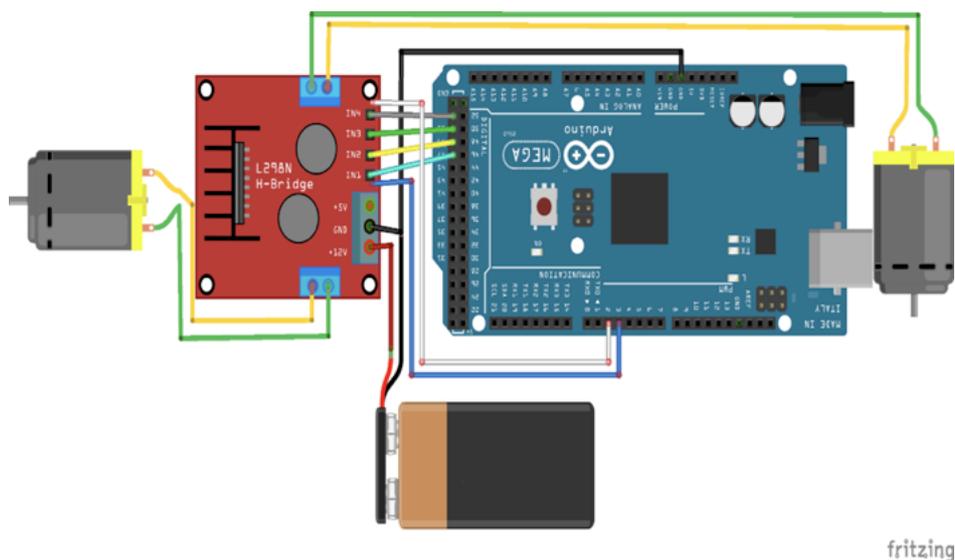


Figura 9. Diagrama de ligação do Arduino, Ponte H e bateria. Fonte: Autor.

A Figura 10 apresenta o protótipo utilizando como base uma espuma anti-chamas e estática, utilizada em toners e outros componentes eletrônicos. A espuma foi colocada sobre o chassi e sobre ela o Raspberry PI e o carregador portátil, foi utilizado braçadeiras para prender tudo ao chassi do protótipo. Essa medida foi tomada para evitar cortes na carenagem do carrinho, tornando-o hábil a executar os testes necessários, sem ser alterada de forma drástica e até mesmo errada a sua carenagem.

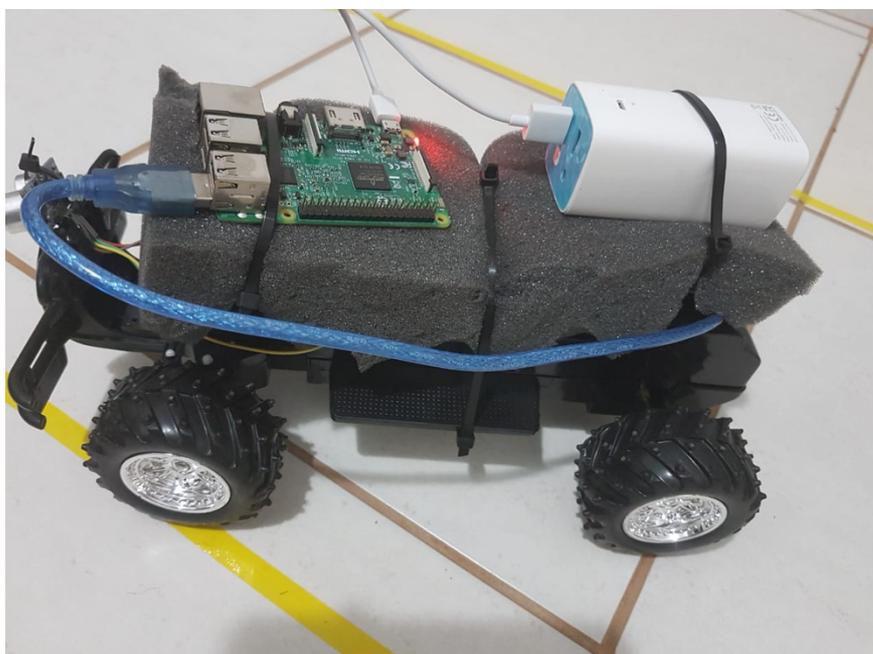


Figura 10. Protótipo com base de espuma, Raspberry PI e carregador portátil. Fonte: Autor.

A Figura 11 representa a ligação entre o Raspberry PI e o Arduino, ela foi efetuada utilizando um cabo USB A x USB B, e a sua entrada micro USB que é responsável por fornecer energia ao Raspberry PI, foi ligado um cabo Micro USB B x USB A a uma das saídas do carregador portátil.

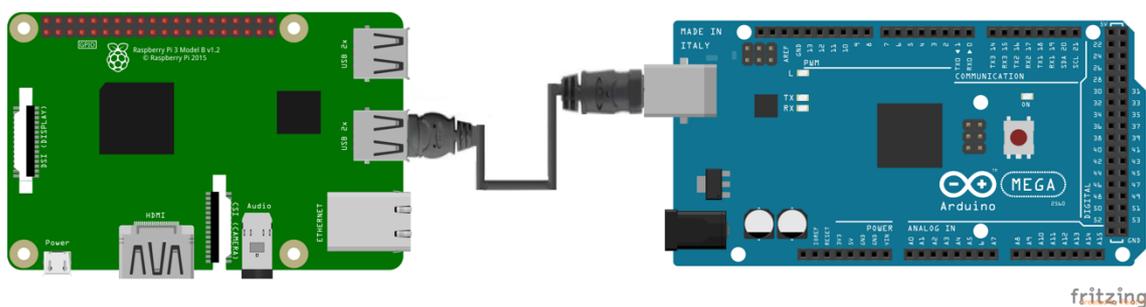


Figura 11. Diagrama de ligação entre Raspberry PI e Arduino. Fonte: Autor.

3.3. Software

Foram efetuadas várias pesquisas sobre inúmeros meios de desenvolver o aplicativo que a ser executado no Raspberry PI, e com o advento da internet das coisas, muitas tecnologias estão surgindo com o intuito de facilitar o desenvolvimento de aplicações voltadas a automação residencial, industrial, robótica e outras. Neste contexto foi verificado o protocolo Firmata e seu firmware para Arduino.

Como é descrito por [Firmata firmware for Arduino, 2018], Firmata é um protocolo que permite que seja efetuada a comunicação direta entre o Arduino e um computador host. No caso do protótipo é a comunicação entre Raspberry PI e Arduino.

No projeto poderia ser desenvolvidas duas aplicações, uma para o Arduino utilizando C, e outra em qualquer linguagem que fosse possível executar no Raspberry PI, e que fosse capaz de fornecer acesso ao usuário à uma aplicação web, capaz de permitir o controle do carrinho.

A escolha por utilizar o NodeJs, e não desenvolver uma aplicação em “C” para o Arduino, e para aplicação outra linguagem, baseou-se na grande adoção do NodeJs em projetos de internet das coisas, na sua facilidade de aprendizagem, e agilidade no desenvolvimento de aplicações. A possibilidade de colocar em pratica o uso do módulo Johnny-Five na robótica, uma de suas áreas propostas, e o desenvolvimento de uma única aplicação, responsável pelo controle e comunicação com o Arduino, e ao mesmo tempo servidora da aplicação web, contribuíram para essa escolha.

NodeJs é uma tecnologia assíncrona para o desenvolvimento de aplicações escaláveis utilizando JavaScript [Node.js, 2018]. Tem como conceito que nenhuma requisição bloqueia a thread principal da aplicação, ficando ela a cargo de delegar as requisições solicitadas ao servidor. Para facilitar seu desenvolvimento conta com uma grande diversidade de módulos disponibilizados na plataforma de gerenciamento de pacotes NPM e YARN, por padrão o NodeJs utiliza o NPM para gerenciar seus pacotes, já vem instalado o NPM como padrão.

No projeto foram utilizados os módulos Express, SocketIO e Johnny-Five.

Express é um framework para desenvolvimento Node.js, minimalista, flexível, que fornece um conjunto robusto de recursos para aplicativos web e móvel [ExpressJs, 2018].

SocketIO estende o conceito do protocolo do WebSockets, implementando a comunicação por socket bidirecional baseada na recepção e emissão de eventos em Real-Time [Socket.IO, 2018].

Johnny-Five é um framework para robótica e internet das coisas, utiliza do protocolo Firmata de comunicação com diversas plataformas, como o Arduino e Raspberry [Johnny-Five, 2018].

A aplicação em NodeJs tem como conceito ser um servidor web, dispensando a instalação e configuração de um servidor para utilizar a página web desenvolvida. A utilização do módulo Express facilita a criação de rotas e configuração do servidor.

Foi configurado o servidor para utilizar a porta 8080 e uma rota direta, a raiz, sendo acessada no browser utilizando o endereço de rede IP do Raspberry PI mais a porta, exemplo: "http://10.0.0.251:8080".

O módulo SocketIO é carregado e configurado para utilizar o módulo Express, assim ambos utilizam a mesma porta do servidor, e foi desenvolvido um controller que gerencia os eventos de recepção dos comandos de controle do carrinho.

A comunicação entre o Johnny-Five e o Arduino é feita utilizando o protocolo Firmata, para que o Arduino esteja apto a funcionar com o Johnny-Five, foi implementado o firmware "StandardFirmataPlus", e seu envio foi feito utilizando a IDE do Arduino.

O módulo Johnny-Five é carregado já na inicialização da aplicação, criando um objeto do tipo Board do módulo, o objeto foi configurado sem uma porta padrão, assim ele ativa o monitoramento das portas USBs do Raspberry PI, aguardando a comunicação com o Arduino que contenha implementado um firmware com protocolo Firmata, assim torna possível ligar o Arduino em qualquer USB do Raspberry PI.

Ao efetuar a conexão com o Arduino é disparado o evento de "ready" do objeto do tipo Board, neste evento é carregado o controle dos motores da Ponte H, criando um controlador com um objeto do tipo Motor disponibilizado pelo módulo Johnny-Five, parametrizando o objeto com as portas que fazem a conexão entre o Arduino e a Ponte H, seguindo o padrão definido pelo módulo.

A Figura 12 apresenta a página web desenvolvida, ela é de fácil utilização, permitindo que sejam utilizados os botões direcionais do teclado do computador ou botões pressionados na própria página. Também conta com um botão "Parar Tudo" que permite a parada de todos os controles enviados.



Figura 12. Página html da aplicação no Raspberry PI para controle do protótipo. Fonte: Autor.

A comunicação entre a página Html e o servidor é efetuada usando WebSockets com módulo SocketIO em sua versão cliente. Para efetuar a comunicação com o servidor, são transmitidos os eventos disponibilizados por ele ao clicar nos botões da

página, ou ao pressionar uma das teclas de direção do teclado do computador. É possível utilizar a combinação de teclas do teclado para controlar o veículo na direção desejada, pressionando juntas uma das teclas direcionais esquerda e direita e uma das direcionais para cima e para baixo.

3. Resultados

Após a conclusão do desenvolvimento do projeto, foram efetuados testes de dirigibilidade do veículo, acessando a página web de controle do protótipo, disponibilizada pela aplicação no Raspberry PI. Ligado à rede por um roteador sem fio, foram utilizadas as teclas direcionais do teclado do computador e os botões de controle na página web. Em vista que o protótipo se origina de um carrinho de controle remoto, a sua dirigibilidade se manteve igual e o tempo de resposta também.

A comunicação entre o Raspberry PI e o Arduino através do protocolo Firmata, implementado na aplicação em NodeJs com o módulo Johnny-Five se mostrou estável e confiável.

4. Discussão

O objetivo principal deste artigo foi a elaboração de um carro guiado remotamente em escala reduzida, convertendo um carrinho de controle remoto, para um controlado pelo Raspberry PI.

Um dos desafios encontrados no projeto foi encontrar um carregador portátil que suprisse as especificações do Raspberry PI. Foram necessárias pesquisas em sites de vendas de informática, eletrônicos, e no fórum do site do Raspberry em busca de um modelo. Ao encontrar um possível modelo, foi pesquisado em fornecedores locais de equipamentos de informática, algum que também fornecesse o modelo do carregador e que permitisse que fosse efetuado um teste com o Raspberry, encontrado e testado foi adquirido o carregador modelo Tl-pb10400 da marca TP-Link.

Outra dificuldade encontrada, foi a comunicação entre o Raspberry PI e Arduino, a qual foi superada com pesquisas em projetos de internet das coisas, robótica, projetos similares ao proposto e projetos que efetuavam a comunicação entre ambos. Foi encontrado o módulo Johnny-Five para NodeJs, foram efetuados testes de comunicação e adotado utilização do módulo no projeto, não necessitando de duas aplicações para o controle do protótipo, tornando o desenvolvimento da aplicação mais simplificada.

Os resultados obtidos nos testes e o aprendizado adquirido tornam o projeto proposto finalizado. O projeto cumpriu todos os requisitos propostos, convertendo o carrinho de controle remoto, para um controlado por uma aplicação sendo executada no Raspberry PI instalado no protótipo, não dependendo de processamento externo, somente da configuração de conexão de rede sem fio, podendo ser feita utilizando um roteador ou, configurando a rede sem fio do Raspberry PI para se tornar um ponto de acesso.

A partir desse projeto, podem ser desenvolvidos diversos outros, como um App para dispositivos Android de controle do protótipo. Suporte para controle utilizando o bluetooth do Raspberry PI, sincronizando e configurando as teclas do controle joystick bluetooth. Outro possível projeto seria implementar sensores para a detecção de obstáculos e torná-lo apto a evitar a colisão com esses obstáculos. Tornar o protótipo capaz de se locomover de forma autônoma, desviando de obstáculos. E por fim tornar o protótipo capaz de efetuar manobras para estacionar em vagas de forma autônoma.

4. Referências

Schneider, M. (2005) “Automotive Radar – Status and Trends”, https://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-14581/Paper/5_3.pdf.

- Sabbagh, V. (2009) “Desenvolvimento de um Sistema de Controle para um Veículo Autônomo”, http://www.coro.cpdee.ufmg.br/publications/vitor_sabbagh.pdf.
- Seetharaman, G., Lakhota, A., e Blasch, E. (2006). “Unmanned vehicles come of age: The Darpa grand challenge”, In: Computer, 39(12), p. 26–29.
- Myers, N. (2017). “RC to Rπ Car Conversion”, <http://commons.lib.niu.edu/handle/10843/1748>.
- Raspberry PI. Especificações Técnicas do Produto. Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. Acesso em: 24 de agosto de 2018.
- Arduino. Especificações Técnicas do Produto. Disponível em: <https://store.arduino.cc/usa/arduino-mega-2560-rev3>. Acesso em: 24 de agosto de 2018.
- Hirzel, Timothy. PWM. Disponível em: <https://www.arduino.cc/en/Tutorial/PWM>. Acesso em: 24 de agosto de 2018.
- Reis, Fábio. Curso de Eletrônica. Como funciona uma Ponte H – Controle direcional de motores DC, set. 2017. Disponível em: <http://www.bosontreinamentos.com.br/eletronica/curso-de-eletronica/como-funciona-uma-ponte-h-controle-direcional-de-motores-dc/>. Acesso em: 24 de agosto de 2018.
- TP-Link. Características de Hardware. Disponível em: https://www.tp-link.com/br/products/details/cat-5689_TL-PB10400.html. Acesso em: 24 de agosto de 2018.
- Firmata firmware for Arduino. Descrição do protocolo e firmware. Disponível em: <https://github.com/firmata/arduino>. Acesso em: 24 de agosto de 2018.
- Node.js. Projeto Node.js. Disponível em: <http://expressjs.com/pt-br/>. Acesso em: 24 de agosto de 2018.
- ExpressJs. Projeto ExpressJs. Disponível em: <https://nodejs.org/en/about/>. Acesso em: 24 de agosto de 2018.
- Socket.IO. Projeto Socket.IO. Disponível em: <https://socket.io/>. Acesso em: 24 de agosto de 2018.
- Johnny-Five. Projeto Johnny-Five. Disponível em: <http://johnny-five.io/>. Acesso em: 24 de agosto de 2018.
- Baú da Eletrônica. Descrição do produto Driver Motor ponte-H - L298N. Disponível em: <http://www.baudaeletronica.com.br/driver-motor-ponte-h-l298n.html>. Acesso em: 24 de agosto de 2018.