CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA
FATEC SANTO ANDRÉ
Tecnologia em Eletrônica Automotiva

José Rodrigues Moreira
Marcelo Pesci

CARRO EXPLORADOR AUTÔNOMO

Santo André – São Paulo
2017

José Rodrigues Moreira

Marcelo Pesci

CARRO EXPLORADOR AUTÔNOMO

Trabalho de conclusão de curso da Faculdade de Tecnologia - Fatec Santo André, referente ao curso de Eletrônica Automotiva, orientado pelo Professor Cléber Willian Gomes.

Santo André – São Paulo

2017

Moreira, José Rodrigues
Pesci, Marcelo

Carro Explorador autônomo
José Rodrigues Moreira, Marcelo Pesci
Santo André, 2017– f: 51 il.

Trabalho de conclusão de curso - FATEC- Santo André.
Curso de Eletrônica Automotiva, 2017.

Orientador: Cleber Willian Gomes

1.Explorador Autônomo 2. Desviar de obstáculos
I. Moreira, José Rodrigues II. Pesci, Marcelo

LISTA DE PRESENÇA

Santo André, 21 de Dezembro de 2017

LISTA DE PRESENÇA REFERENTE À APRESENTAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO COM O TEMA: **"CARRO EXPLORADOR AUTÔNOMO"**

DOS ALUNOS DO 6º SEMESTRE DESTA U.E.

BANCA

PRESIDENTE:
PROF. CLEBER WILLIAN GOMES _____

MEMBROS:
PROF. ROBERTO BORTOLUSSI _____

PROF. FERNANDO GARUP DALBO _____

ALUNO:
JOSÉ RODRIGUES MOREIRA _____

MARCELO PESCI _____

AGRADECIMENTOS

Agradecemos a FATEC pela oportunidade e pelos ensinamentos, a todos os professores que fizeram parte dessa formação e ao nosso orientador, aos familiares pela paciência e atenção durante todos os anos letivos. Enfim, a todos os amigos que direta ou indiretamente incentivaram e acreditaram na execução desse projeto.

RESUMO

Carro Explorador Autônomo (CEA) visa proporcionar aos condutores maior segurança e comodidade ao trafegarem em terrenos com diferentes tipos de obstáculos como: carros, pedestres, arvores, muros, animais e outros. Cabe ao motorista definir quando da partida do veículo e a sua parada a qualquer momento do trajeto ou percurso, ficando por conta do CEA reconhecer e fazer o desvio dos diferentes obstáculos. O projeto consta de um modelo em miniatura adaptado com sensores eletrônicos localizado na parte dianteira do veículo com objetivo de informar a uma Central da aproximação de um determinado obstáculo. Será utilizada em nosso projeto uma placa eletrônica com um micro controlador para fazer esse gerenciamento, onde uma Central recebe os dados fornecidos pelos sensores espalhados pelo veículo que identifica e interpreta a situação naquele instante e enviando comandos a seus atuadores interligados, a seleção da direção a ser tomada será feita tomando como referência o maior espaço para que o carro passe com segurança e com objetivo de tomar uma decisão de qual direção deve seguir, deve manter uma velocidade máxima de 10 km/h.

Palavras-chaves: Explorador autônomo, Desvio de obstáculos.

ABSTRACT

Car Autonomous Explorer (CEA) aims to provide drivers with greater safety and convenience when traveling on terrains with different types of obstacles such as: cars, pedestrians, trees, walls, animals and others. It is the responsibility of the driver to determine when the vehicle starts and stops at any point along the route or course, and CEA recognizes and diverts the different obstacles. The design consists of a miniature model adapted with electronic sensors located in the front of the vehicle with the purpose of informing a Central of the approximation of a certain obstacle. An electronic board with a microcontroller will be used in our project to do this management, where a central receives the data provided by the sensors scattered by the vehicle that identifies and interprets the situation at that moment and sending commands to its interconnected actuators, selecting the direction to being taken will be made taking as reference the largest space for the car to pass safely and in order to make a decision of which direction to follow, it must maintain a top speed of 10 km / h.

Key words: Car Autonomous explorer, Deviation of obstacles.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE APENDICE

LISTA DE SIGLAS E ABREVIAÇÕES

A – Ampere
AC – Corrente Alternada
AVs – Autonomous Vehicles
CA – Corrente Alternada
CC – Corrente contínua
CEA – Carro Explorador Autônomo
CI – Circuito Impresso
DC – Corrente Continua
EEPROM - Electrically Erasable Programmable Read Only Memory
GND – Terra (Graduated Neutral Density Filter)
GPRS – General Packet Radio Service (Serviços Gerais de Pacote por Rádio)
GSM – Groupe Special Mobile (Sistema Global para Comunicações Móveis)
Hz – Hertz
IDE – do inglês *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado
IN – Polegada
KB – Quilo byte
KHZ – Quilohertz
mA – Miliampere
ms – Milissegundos
n – Número de espiras
NASA – Sigla em inglês de National Aeronautics and Space Administration – (Administração Nacional da Aeronáutica e Espaço)
OMS - Organização Mundial de Saúde
IDE – do inglês *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado
PWM – Modulação por largura de pulso(Pulse Width Modulation)
PWR – Power
SRAM - Static Random Access Memory (Memória estática de acesso aleatório)
USB – Universal Serial Bus (Porta Universal)
Vcc – Tensão contínua

SUMÁRIO

1. Introdução

Veículos autônomos podem visualizar 360° a sua volta, são capazes de analisar situações de perigo e reconhecer vários tipos de obstáculos com maior rapidez e segurança se comparado a um ser humano. Por ser autônomo não teria problema com cansaço, distração e com consumo de bebidas alcoólicas durante sua condução, porem um condutor seria passível a esses tipos de ocorrências.

> Segundo a Organização Mundial de Saúde (OMS), cerca de 1,25 milhão de pessoas morrem em acidentes de carro todos os anos no mundo. Os carros autônomos podem ajudar a mudar essa realidade! Além disso, eles podem gerar outros benefícios à sociedade, como promover um tráfego mais eficiente, a partir da comunicação entre os veículos e a criação de rotas comuns, tecnologias que podem revolucionar a forma de viajar. (UDACITY, junho 2017).

Um veículo capaz dirigir sozinho, tomar decisões quanto a direção a seguir ou parar. Uma central recebe dados fornecidos por sensores que identificam os possíveis obstáculos no caminho e em seguida interpreta a situação e escolher a melhor forma de agir enviando comandos a seus atuadores interligados a esta central.

Para fazer essa integração entre sensores e atuadores foi utilizada uma placa eletrônica com um micro controlador para gerenciar todos os dispositivos acoplados a ele, uma placa Arduino UNO. Essa placa permite que seja acoplada a ela diversos sensores e atuadores e ser controlada remotamente, pode ser configurada de várias formas através de programação do software.

A Plataforma Arduino possibilita conectar todos os periféricos necessários para a construção do nosso Carro Explorador Autônomo juntamente com o sistema de controle a ser implementado. A adaptação e eficácia necessita tempo de resposta aceitável para o correto processamento dos dados dos sensores para que o desvio do obstáculo seja feito de maneira segura e eficiente. Assim este trabalho de conclusão de curso representa em parte nosso aprendizado adquirido durantes esses anos letivos no curso de Tecnologia em Eletrônica Automotiva, Fatec Santo André.

1.1.Objetivo

O de desenvolver um sistema semelhante aos já existente no mercado automotivo, utilizando-se de ferramentas e aprendizado adquirido durante curso, chamado Carro Explorador Autônomo. O embasamento teórico para implementação do projeto, os procedimentos metodológicos adotados e a justificativa para elaborar e realizar a montagem desta plataforma robótica. A instalação de sensores ultrassônicos na parte dianteira do veículo, montagem de motores DC para fazer a movimentação para frente e a ré, servo motor para atuar junto a direção virando para o lado esquerdo e direito e desenvolver uma programação para placa Arduino para reconhecimento de obstáculos e tomada de decisões autônoma.

1.2. Motivação

Com a implantação desses sistemas nos veículos, além de apresentar conforto e comodidade, pode-se dizer que faz parte de um sistema de segurança, ao se deslocar de um ponto de origem até um determinado local desviando de obstáculos e se deslocando em um terreno irregular, pois ao se aproximar de algum tipo de obstáculo, ele irá tomar a decisão de desviar ou parar o veículo com distância segura sem que ocorra colisão, devido aos sensores instalados. O que o torna um sistema altamente confiável.

2. Desenvolvimento

2.1.Carro Autônomo

Conforme Bayard (2017), o carro autônomo conhecido pela sigla inglês AVs (Autonomous Vehicles) é hoje o centro de pesquisas e desenvolvimento da atualidade no mundo, visa mudança dos atuais motores a combustão para motores mais modernos e elétricos, tornando mais eficientes e com capacidade de autonomia maior. Os veículos atuais já contam com assistência para estacionar, cambio automáticos entre outras funções que já não exige do motorista tal tarefa. Não só as grandes empresas montadoras de automóveis estão nessa briga para dominar as tecnologias necessárias para o

desenvolvimento de AVs, as gigantes da internet como Google e Apple também estão na disputa por este mercado em ascensão. Todas elas trabalham arduamente em volta de três tecnologias que fazem um carro ser autônomo (figura 01):

- Sensores e atuadores*;*
- Conectividade*;*
- Algoritmos de controle*;*

Figura 01: Visão de um carro autônomo

Fonte: Futurecom

2.2. Sensores e Atuadores

Os sensores são responsáveis por escanear todo o ambiente ao redor do carro e prover informações necessárias para a navegação, aceleração e frenagem corretas. Radares, sensores ultrassônicos, câmeras e módulos de navegação inercial são alguns dos sensores mais importantes para dar percepção situacional a um veículo.

Os atuadores consistem em sistemas de servomecanismo e motores elétricos responsáveis pelo sistema de direção, aceleração, frenagem, e qualquer ação que normalmente seria tomada pelo motorista.

2.3. Conectividade

A conectividade diz respeito ao acesso às informações sobre o clima, o tráfego de veículos, condições do asfalto local, comunicação com outros carros, infraestrutura nas rodovias e mapas. Todas essas informações são usadas para otimizar o sistema de navegação e maximizar a dinâmica de frenagem e aceleração do carro. É também o calcanhar de Aquiles das pesquisas atuais. A conectividade é o flanco pelo qual veículos com sistemas de segurança vulneráveis podem ser hackeados. Essa é uma das principais preocupações dos projetistas.

As tecnologias usadas para conectar os carros são variadas. Desde de redes Wifi, bluetooth, 4G, 5G até GPRS e GSM. Vários são os protocolos de comunicação usados para obter informações para o carro.

2.4. Algoritmos de controle

A conectividade e os sensores provêm as informações necessárias para o carro autônomo, mas para tomar as decisões corretas e acionar os atuadores na medida certa, é preciso um software que processe a base de informações e produza uma saída como resultado. Esses softwares são os algoritmos de controle responsáveis por processar toda a base de dados que os sensores enviam para central, a função da conectividade é captar esses dados e acionar os atuadores de forma segura e na medida correta.

3. Metodologia

3.1. Materiais e métodos

Serão apresentados todos os detalhes envolvidos na construção do Carro Explorador Autônomo, desde a plataforma escolhida até as programações

implementadas, descrever os materiais e a metodologia aplicados nesse projeto, os conceitos por traz da criação deste veículo, a implementação do software de controle com código fonte e os resultados obtidos.

3.2. Carro Explorador Autônomo

O Carro Explorador Autônomo será composto por um modelo de veículo em miniatura, uma placa Arduino UNO, motores DC, placa controle Shields, sensor ultrassônico HC-SR04, servo motor e as baterias necessárias. A seguir são apresentadas e descritas com detalhes de funcionalidades de cada componente e a plataforma utilizada. A (figura 02) mostra um Veículo Lunar utilizado pela NASA para exploração da superfície da Lua.

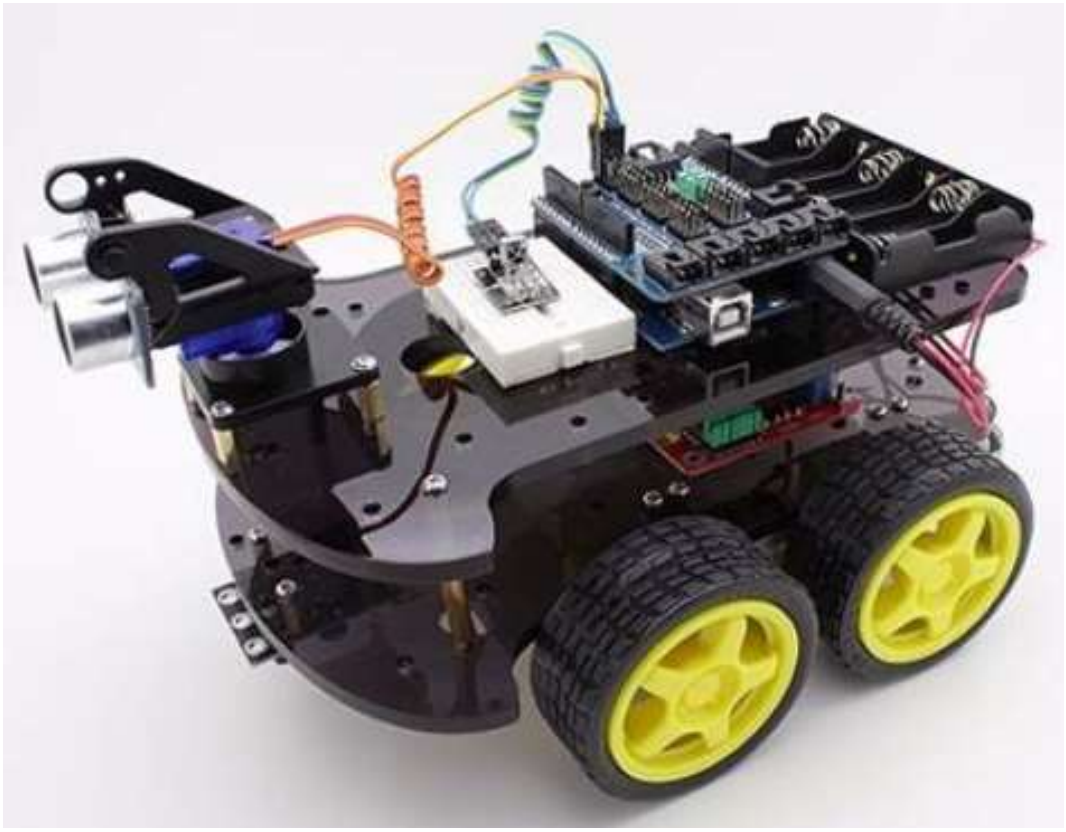
Figura 02: Veiculo Explorador Lunar

Fonte: Jet Propulsion Laboratory, em Pasadena, Califórnia, sede da NASA, Washington, D.C. - julho de 2012.

3.3. Plataforma de Montagem

A escolha dessa plataforma Arduino com micro controlador (figura 03) teve as seguintes características: fácil programação, diversos acessórios, seu software é livre e não requer licença, disponibilidade de diversas bibliotecas para o uso desta plataforma, muita informação em livros, tutoriais e o site Arduino, o que nos ajuda no desenvolvimento do projeto.

Figura 03: Plataforma montagem Robô Arduino



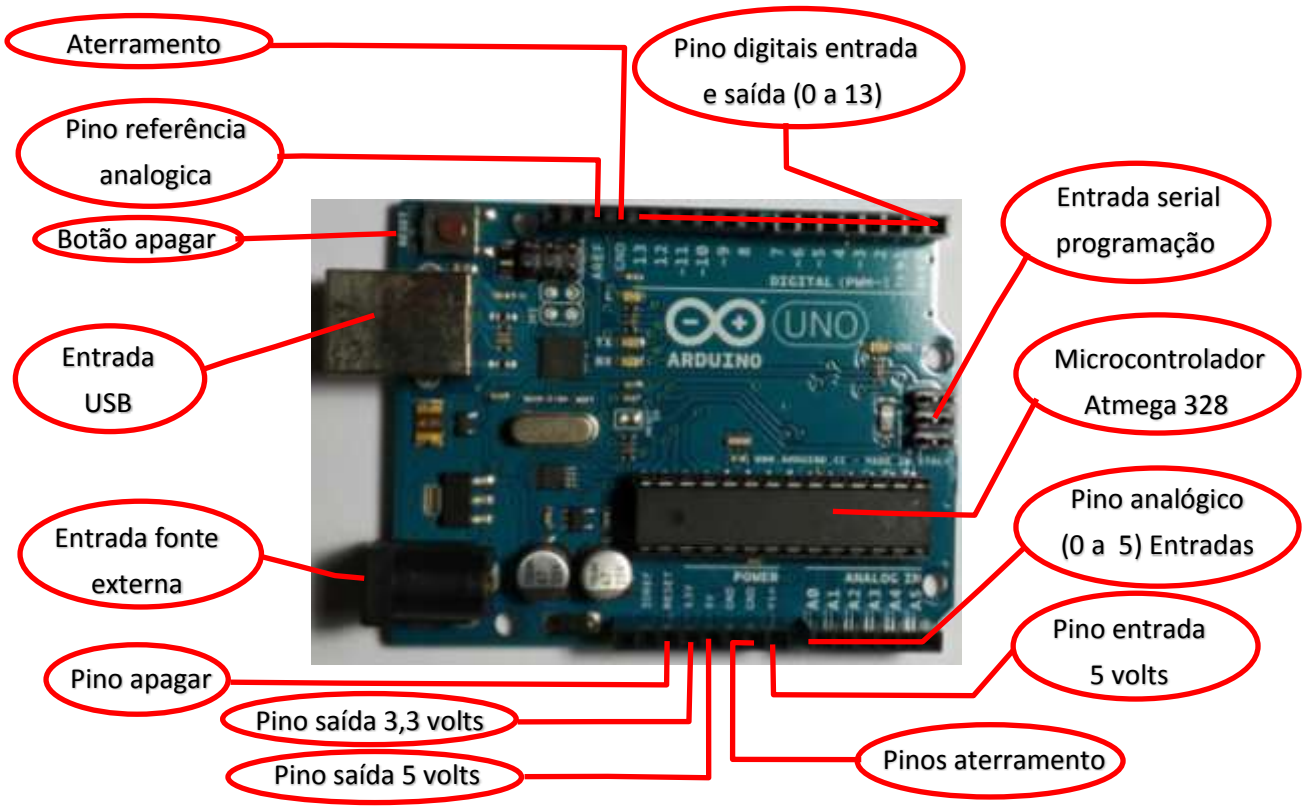
Fonte: Melinterest

3.4. Plataforma Arduino Uno

Segundo Arduino e Cia (2017) o Arduino Uno (figuras 04 e 05) é uma plataforma de computação open-source (Código aberto) baseada em uma simples placa com entradas e saídas tanto digitais como analógicas. Possui um ambiente próprio de desenvolvimento que implementa a Linguagem C. O Arduino pode ser usado para desenvolver objetos interativos autônomos e também ser conectado a um software em um computador. O Ambiente de desenvolvimento (IDE) open-source pode ser obtido gratuitamente (atualmente disponível para Mac OS X, Windows e Linux).

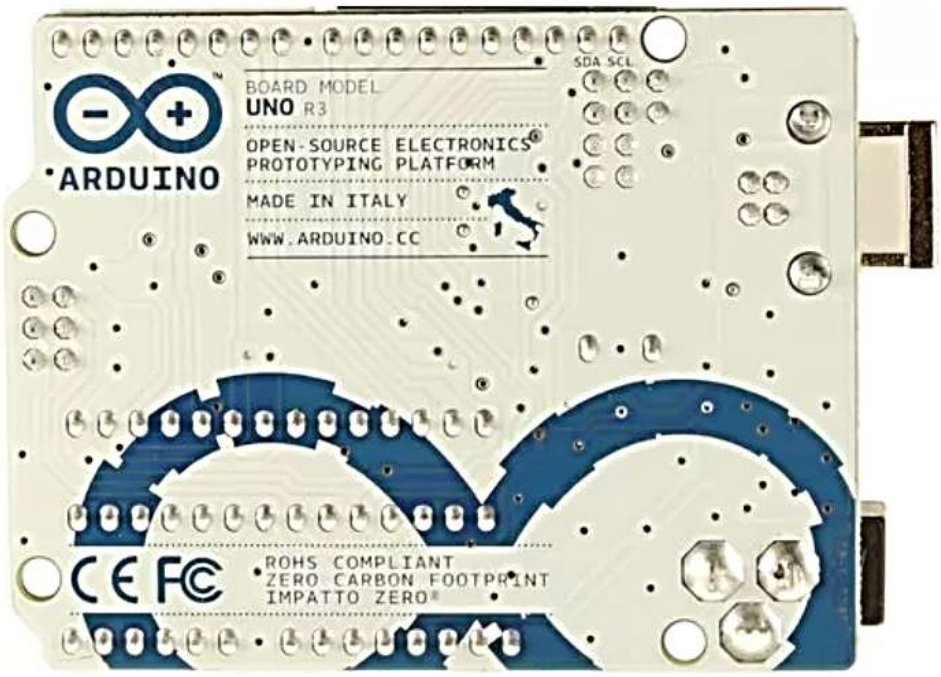
Utilizá-lo é muito fácil, basta o usuário possuir o seu cabo USB, e o software para compilação de códigos. Deste modo, quando conectado em um computador poderá ser facilmente programado e podendo ser utilizado junto de acessórios como shields e módulos para aumentar sua funcionalidade.

Figura 04: Visão frontal Arduino Uno



Fonte: Autores

Figura 05: Visão traseira Arduino Uno



Fonte: Autores

3.5. Características

Arduino Uno (2017), a placa Arduino pode ser alimentada tanto pela porta USB quanto por uma fonte externa de alimentação, sendo feito uma seleção automática do tipo de alimentação. A alimentação externa pode ser feita de duas formas, através de bateria ou adaptador AC-DC.

> Na alimentação com fonte externa a placa pode suportar de 6 a 20 volts, porém recomenda-se uma alimentação entre 7 e 12 volts, para evitar instabilidade na placa ao fornecer menos energia e evitar queimar o dispositivo fornecendo energia acima do recomendado, o micro controlador ATmega328 usado pelo Arduino Uno possui 32 KB de memória, reservando desse total 0,5 KB para o bootloader (gerenciador de boot ou pequeno software instalado no micro controlador responsável por gerenciar a execução dos programas em determinados ciclos de tempo), possui também 2 kB de SRAM e 1 kB de memória EEPROM, esta última pode ser manipulada usando a biblioteca de mesmo nome. (ARDUINO, 2017).

De acordo com o Datasheet Arduino Uno a placa com micro controlador Atmega328. Possui 14 entradas/saídas digitais (das quais 6 podem ser usadas como saídas

PWM), 6 entradas analógicas, um cristal oscilador de 16MHz, conexão USB, uma entrada para fonte, soquetes para ICSP e um botão de reset. Simplesmente conecte-a a um computador com o cabo USB ou ligue a placa com uma DC (ou bateria). O Uno seleciona automaticamente a fonte de alimentação (USB ou fonte externa). Esta placa já vem pronta e testada com o micro controlador ATMega328 pré-carregado com "bootloader". A placa Uno se diferencia das outras por não utilizar o chip da FTDI USB-to-Serial. Ao invés deste chip, um Atmega8U2 já programado faz a função de converter os dados da USB para Serial. (ARDUINO, 2017).

Quadro 01: Características

Tamanho:	5,3cm x 6,8cm x 1,0cm
Micro controlador:	ATmega328
Tensão de operação:	5V
Tensão de entrada (recomendada):	7-12V
Tensão de entrada (limites):	6-20V
Pinos de entrada/saída (I/O) digitais:	14 (dos quais 6 podem ser saídas PWM)
Pinos de entrada analógicas:	6
Corrente DC por pino I/O:	40mA
Corrente DC para pino de 3,3V:	50mA
Memória Flash:	32KB (dos quais, 0,5KB são usados pelo *bootloader*)
SRAM:	2KB
EEPROM:	1KB
Velocidade de Clock:	16MHz
Temperatura de operação: de 10º a 60º	

Fonte: Datasheet Arduino Uno

3.6. Motores DC

Segundo Kalatec (2017). Motores de corrente continua (CC) ou motores (DC – Direct Current) (figura 06 e 07) são dispositivos que operam aproveitando todas as forças de atração e repulsão geradas por eletroímãs e imãs permanentes. Existem vários tipos desses motores no mercado, tais como os feitos de imãs permanentes com ou sem escovas ou os de relutância variável. Esses já podem ser encontrados numa grande faixa de tensões nominais, tipicamente entre 1,5 a 48 volts. Esse tipo de produto possui vários tamanhos e tensões de trabalho.

Assim, o tamanho de cada motor está diretamente conectado a sua potência, ou seja, quanto maior, mais potente ele será.

Figura 06: Motor DC para Arduino

Fonte: Autores

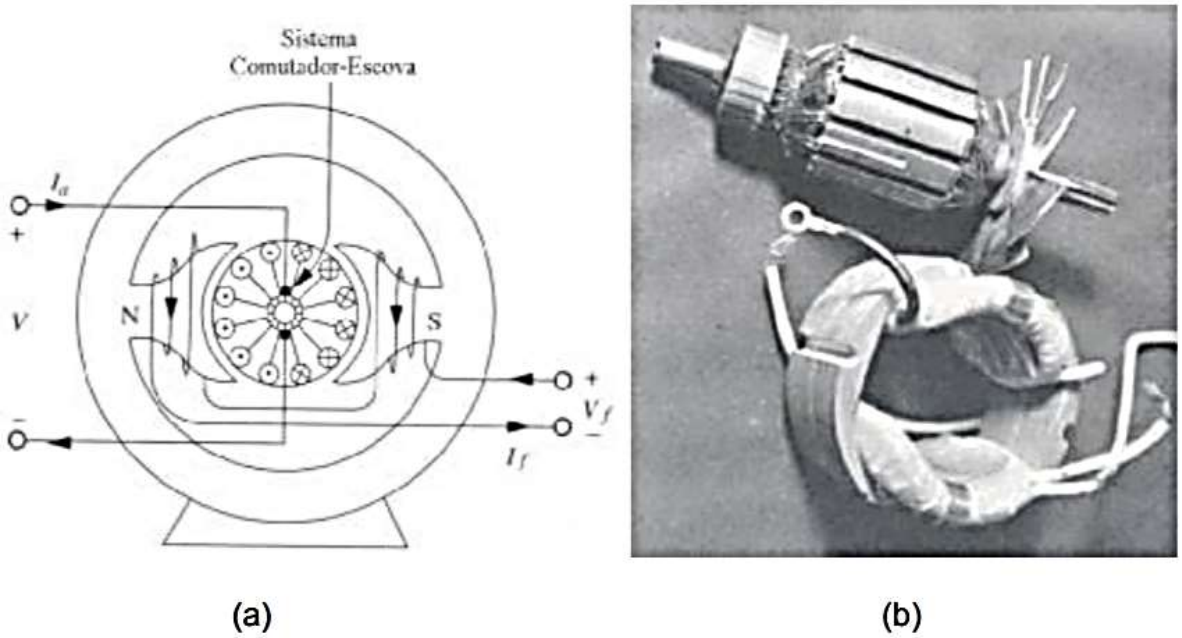
Figura 07: Motor DC acoplado a roda

Fonte: Autores

Siemens Mario Loureiro (2017). O motor de corrente contínua é composto de duas estruturas magnéticas: estator (enrolamento de campo ou ímã permanente) e o rotor (enrolamento de armadura). O estator é composto de uma estrutura ferromagnética com polos salientes aos quais são enroladas as bobinas que formam o campo, ou de um ímã permanente.

A (figura 08) mostra o desenho de um motor CC de 2 polos com enrolamento de campo.

Figura 08: desenho (a) e foto (b) de um motor CC de 2 polos



Fonte: Siemens *LTDA Unidade Automação e Controle*

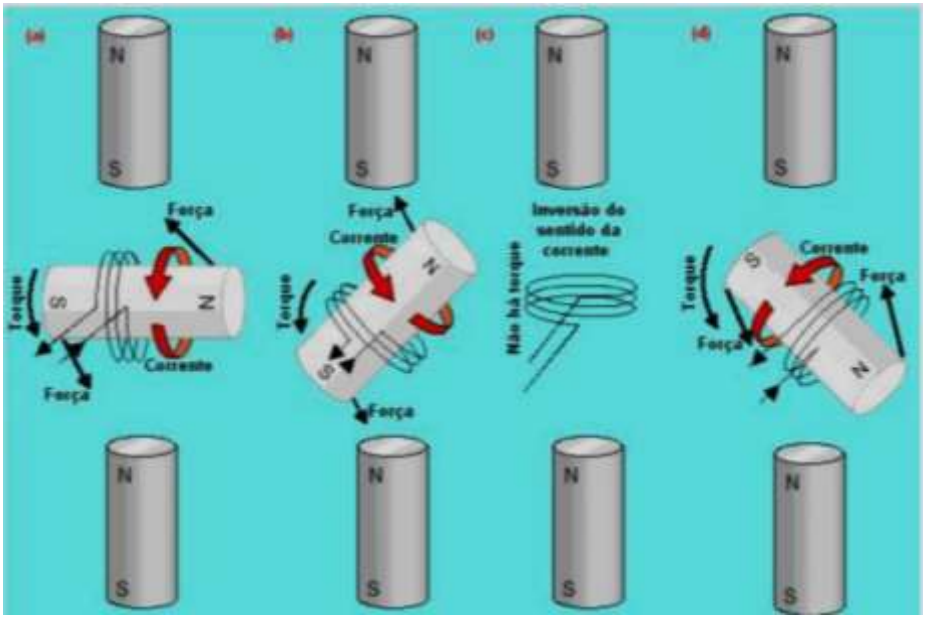
Siemens Mario Loureiro (2017). O rotor é um eletroímã constituído de um núcleo de ferro com enrolamentos em sua superfície que são alimentados por um sistema mecânico de comutação (figura 09). Esse sistema é formado por um comutador, solidário ao eixo do rotor, que possui uma superfície cilíndrica com diversas lâminas às quais são conectados os enrolamentos do rotor; e por escovas fixas, que exercem pressão sobre o comutador e que são ligadas aos terminais de alimentação. O propósito do comutador é o de inverter a corrente na fase de rotação apropriada de forma a que o conjugado desenvolvido seja sempre na mesma direção. Os enrolamentos do rotor compreendem bobinas de (n) espiras. Os dois lados de cada enrolamento são inseridos em sulcos com espaçamento igual ao da distância entre dois polos do estator, de modo que quando os condutores de um lado estão sob o polo norte, os condutores do outro devem estar sob o polo sul. As bobinas são conectadas em série através das lâminas do comutador, com o fim da última conectado ao início da primeira, de modo que o enrolamento não tenha um ponto específico.

Figura 09: Motor Sistema de Comutação

Fonte: Siemens *LTDA Unidade Automação e Controle*

Todos os motores de corrente contínua possuem uma estrutura magnética completamente laminada, sendo, portanto, adequados para utilização com conversor CA/CC, e no caso de processos com alta dinâmica, consegue-se uma taxa de aumento da corrente de até 250xIN por segundo. A (figura 10) mostra o funcionamento do motor CC de dois polos (LOUREIRO, 2017).

Figura 10: Funcionamento do motor CC de dois polos

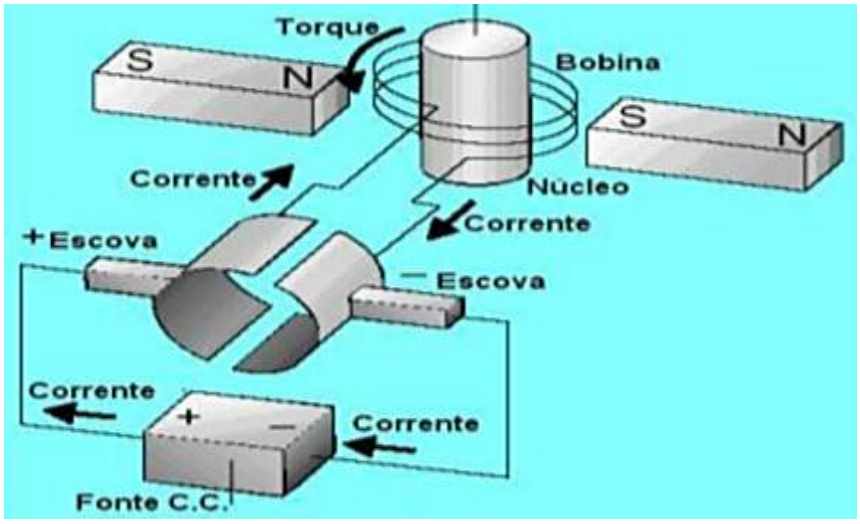


Fonte: Siemens *LTDA Unidade Automação e Controle*

O estator é constituído por ímãs permanentes e o rotor é uma bobina de fio de cobre esmaltado por onde circula uma corrente elétrica. Uma vez que as correntes

elétricas produzem campos magnéticos, essa bobina se comporta como um ímã permanente, com seus polos N (norte) e S (sul) como mostrados na (figura 10). Como os polos opostos se atraem, a bobina experimenta um torque que age no sentido de girar a bobina no sentido anti-horário (a). A bobina sofre aceleração angular e continua seu giro para a esquerda, como se ilustra em (b). Esse torque continua até que os polos da bobina alcancem os polos opostos dos ímãs fixos (estator). Nessa situação (c) – a bobina girou de 90º – não há torque algum, uma vez que os braços de alavanca são nulos (a direção das forças passa pelo centro de rotação); o rotor está em equilíbrio estável (força resultante nula e torque resultante nulo). Esse é o instante adequado para inverter o sentido da corrente na bobina. Agora os polos de mesmo nome estão muito próximos e a força de repulsão é intensa. Devido à inércia do rotor e como a bobina já apresenta um momento angular “para a esquerda”, ela continua girando no sentido anti-horário (semelhante a uma “inércia de rotação”) e o novo torque (agora propiciado por forças de repulsão), como em (d), colabora para a manutenção e aceleração do movimento de rotação. Mesmo após a bobina ter sido girada de 180º, o movimento continua, a bobina chega na “vertical” – giro de 270º –, o torque novamente se anula, a corrente novamente inverte seu sentido, há um novo torque e a bobina chega novamente à situação (a) – giro de 360º. E o ciclo se repete. Essas atrações e repulsões bem coordenadas é que fazem o rotor girar. A inversão do sentido da corrente (comutação), no momento oportuno, é condição indispensável para a manutenção dos torques ”favoráveis”, os quais garantem o funcionamento dos motores. A comutação (figura 11) consiste na mudança de uma lâmina do comutador, onde as bobinas são ligadas em série, para a próxima. Durante esta comutação a bobina é momentaneamente curto circuitada pelas escovas, o que ajuda a liberar energia a armazenada, antes de a corrente fluir no sentido oposto (LOUREIRO, 2017).

Figura 11: Comutador e escovas

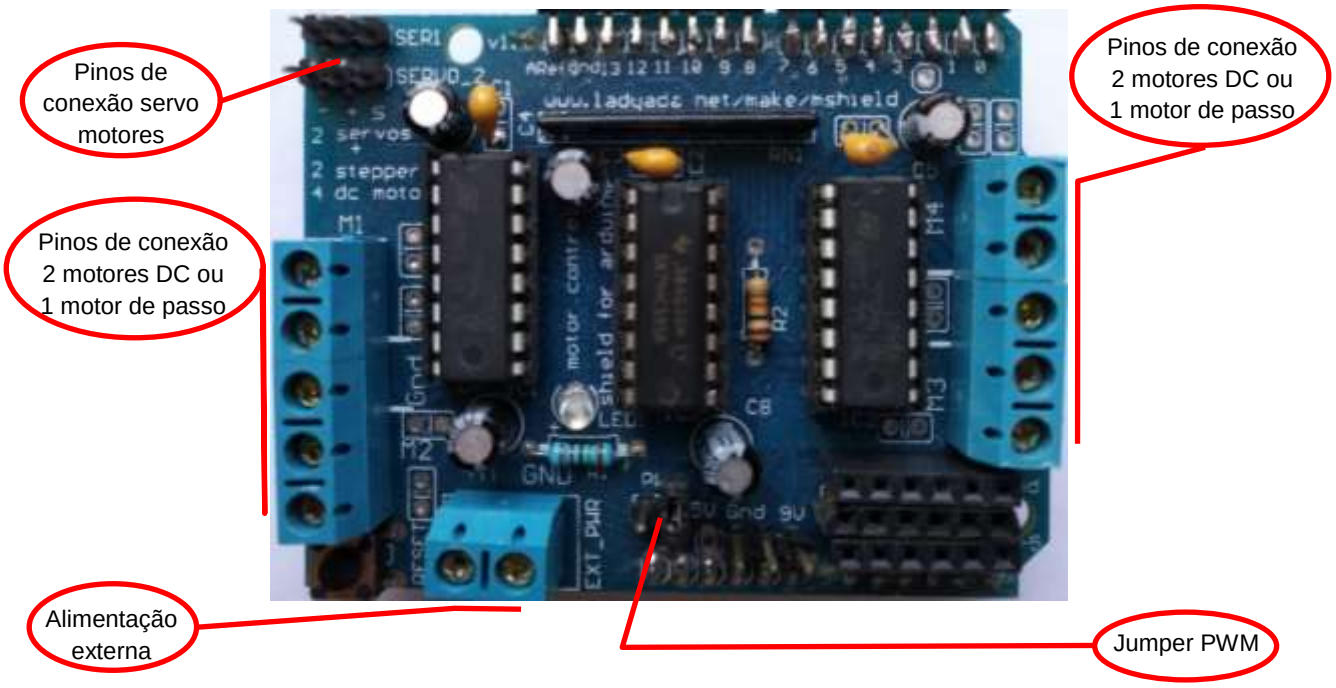


Siemens LTDA Unidade Automação e Controle – Acionamentos e Motores Elétricos
www.siemens.com.br/motores

3.7. Placa SHIELD

Segundo Vidadesilicio (2017) a Placa Shield (figura 12), possui dois chips L293D, cada um composto por 2 pontes H, além de um CI 74HC595. Controla até 4 motores DC, 2 Servos (alimentados por 5 V) ou 2 motores de passo. Corrente máxima de 600 mA, com picos de 1,2 A.

Figura 12: Placa Shield



Fonte: Autores

Nas laterais da placa estão os terminais (com parafusos) para conexão dos motores DC ou motores de passo. Na parte superior esquerda, conectores de 3 pinos permitem a conexão de até 2 servos. Um led na parte central da placa indica não só o funcionamento do shield como também que há alimentação para os motores. A tensão de entrada pode variar de 4,5 à 25 VCC. Para utilização de alimentação externa tem que retirar o jumper PWR. Como a maioria dos shields, você tem à disposição os pinos que sobram quando você não está controlando motores e também alguns pinos que estão sempre disponíveis. (ARDUINO, 2011).

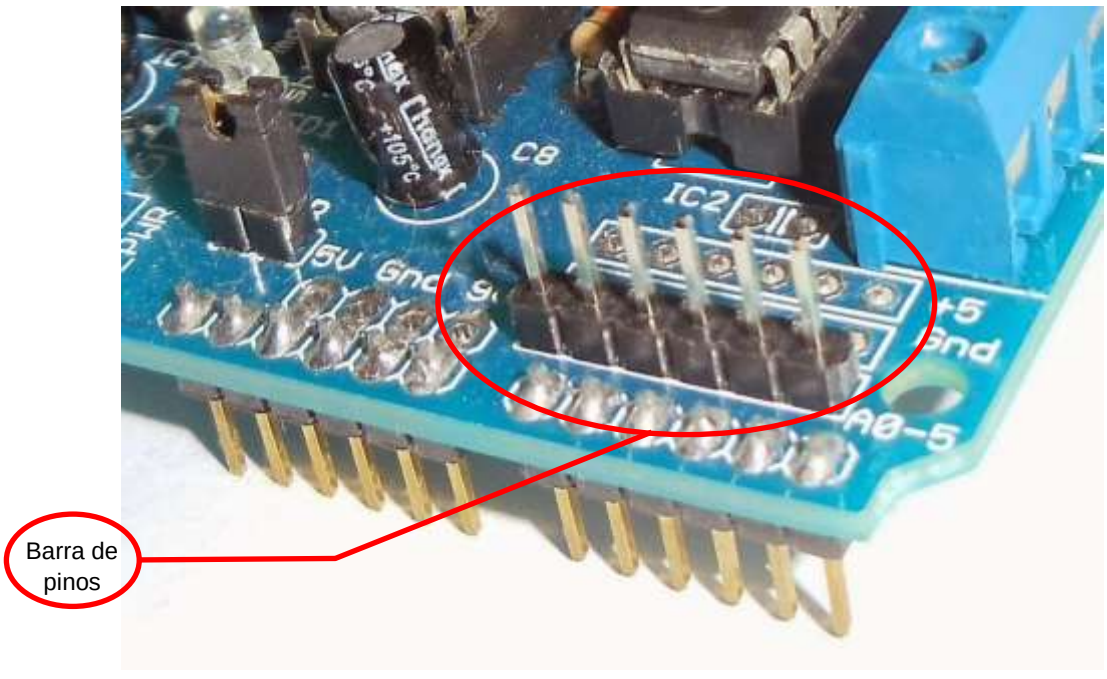
Pinos disponíveis: (Figura 13) os pinos analógicos de A0 a A5, também podem ser utilizados como pinos digitais 14 a 19.

Pinos utilizados para controle de motores DC e motores de passo: 11,3,5 e 6, além dos pinos 4,7, 8 e 12.

Pinos utilizados para controle de servo motores: pinos 9 (servo 1) e 10 (servo 2)

Para utilizar os pinos que sobram, pode-se soldar uma barra de pinos ao shield nos furos correspondentes, como mostra a (figura 13).

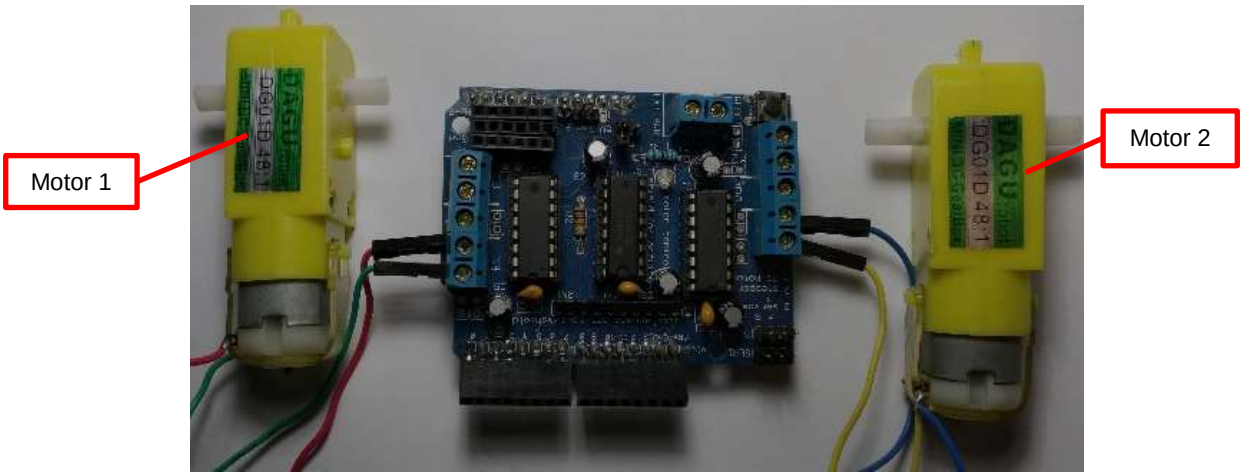
Figura 13: Pinos analógicos A0 /A5



Fonte: Autores

Para ligar 2 motores (figura 14) com o Arduino necessita-se da biblioteca AFMotor. Arquivo encontrado dentro da pasta LIBRARIES da IDE do seu Arduino.

Figura 14: Controlar 2 motores DC de 5 volts, com alimentação externa

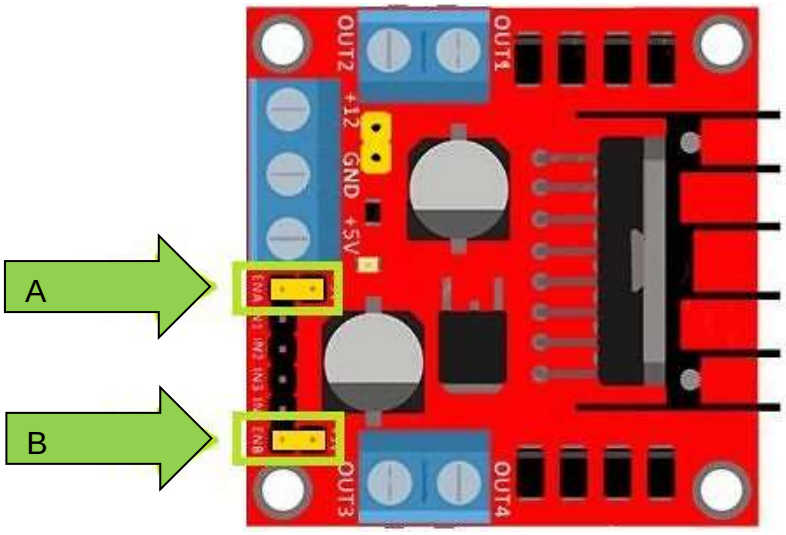


Fonte: Autores

3.8. Ponte H com CI L298N

A Ponte H possui um pino que ativa ou não a ponte H. Caso tenha um sinal de 5 V inserido nele, a ponte entra ligada, caso seja 0 V a ponte estar desligada. Normalmente o Enable A e B (figura 15) fica em curto com um sinal de 5 V da placa através de um jumper (CARDOSO, 2015).

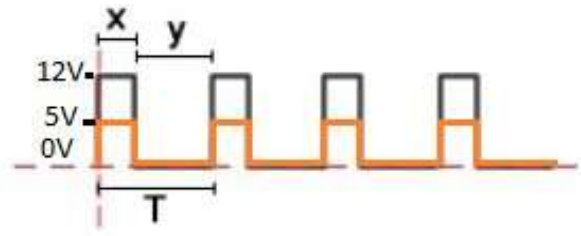
Figura 15: Jumpers Enable A e B



Fonte: Vidadesilicio

Retirando-se esse jumper e inserindo um sinal PWM nessa entrada, modula-se a tensão que é enviada para o motor no mesmo formato. Isso ocorre porque a ponte H só ira “funcionar” enquanto o sinal de enable estive com 5 V (figura 16).

Figura 16: Sinal Enable em curto com sinal PWM



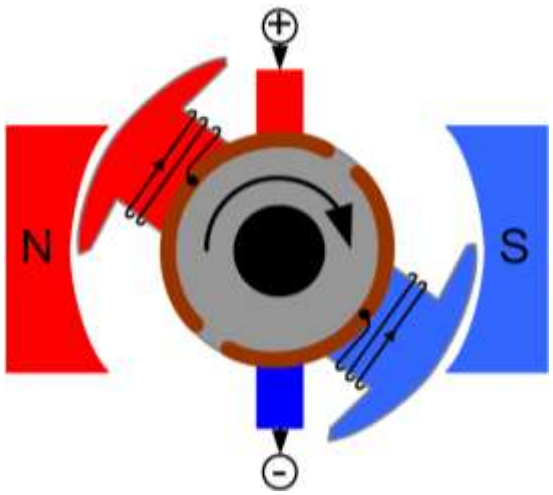
Fonte: Fonte: Vidadesilicio

Com essa modulação, pode-se variar a velocidade do motor através de PWM. Um motor DC gira baseado em campos magnéticos (figura 17) gerados pela corrente que passa em suas bobinas. Para variar a velocidade do motor podemos alterar essa corrente que é diretamente proporcional a tensão sobre elas. Dessa forma, com a mudança da tensão em cima do motor, teremos uma alteração de velocidade usando o Arduino e a Ponte H, (ARDUINO, 2011).

Sinal PWM entrando no Enable A em vermelho (5 V) e a saída para o motor A em preto (12 V), a saída para o motor será um sinal PWM com um Duty Cycle igual ao do Enable e terá tensão média calculada pela seguinte formula.

Vmédio = Vmax (tensão Ponte H) *Duty Cycle (%) (01)

Figura 17: Campo magnético motor DC e sentido giro



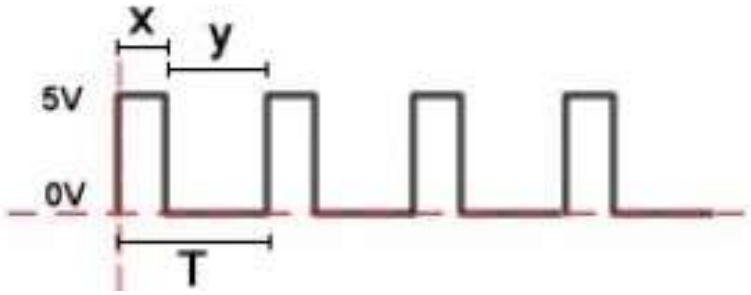
Fonte: Vidadesilicio

3.9. PWM (Pulse Width Modulation)

PWM (Modulação por Largura de Pulso) é uma técnica para obter resultados analógicos por meios digitais. Essa técnica consiste na geração de uma onda quadrada em nível lógico alto e que pode ser controlada o tempo em que a onda permanece em nível lógico alto. Esse tempo é chamado de Duty Cycle (ciclo de trabalho) e sua alteração provoca mudança no valor médio da onda, indo desde 0 V (0 % de Duty Cycle) a 5 V (100 % de Duty Cycle) no caso do Arduino (CARDOSO, 2015).

O duty cycle é a razão do tempo em que o sinal permanece na tensão máxima (5 V no Arduino) sobre o tempo total de oscilação, como está ilustrado na figura abaixo:

Figura 18: Sinal PWM



Fonte: Fonte: Vidadesilicio

Duty Cycle (%) = (x/x+y)*100% = (x/T)*100% (02)

Vmédio = Vmáx*Duty Cycle(%) (03)

O valor do Duty Cycle usado pelo Arduino é um inteiro armazenado em 8 bits (2^8), de forma que seu valor vai de 0 (0 %) a 255 (100 %).

Exemplo: Para um sinal PWM de valor 200 temos:

Se 255 é 100%, 200 é aproximadamente 78,4 %.
Como a tensão máx. de saída do Arduino é 5 V a tensão média do sinal PWM será:

Vmédio = Vmax*Duty Cycle(%)
Vmédio=5*78,4 %
Vmédio=3,92 V

3.10. Sensor HC-SR04

O Sensor Ultrassônico HC-SR04 (figura 19) é um componente muito comum em projetos com Arduino, e permite que sejam feitas leituras de distâncias entre 2 cm e 4 metros, com precisão de 3 mm. Pode ser utilizado simplesmente para medir a distância entre o sensor e um objeto, como para acionar portas do micro controlador, desviar um robô de obstáculos, acionar alarmes, etc. (THOMSEN, 2011).

Figura 19: Sensor HC-SR04

Fonte: Autores

Segundo Hoerpers (2012) sensores são dispositivos que mudam seu estado sob a incidência de alguma grandeza física tornando capaz de perceber as alterações e assim convertendo em sinais elétricos.

O termo ultrassom é empregado para definir ondas acústicas com frequência superior à capacidade audível do ser humano, ou acima 20.000 Hz.

Segundo Webster (1999) os sensores ultrassônicos são dispositivos que, por meio de transdutores elétricos, fazem uso do método de pulso eco para medição de distância. Este método é capaz de detectar a distância de um objeto em função do tempo gasto entre emissão e recepção de uma onda ultrassônica (BASTOS, 1999). O transdutor é um dispositivo que transforma um tipo de energia em outro. Ele pode converter, por exemplo, uma magnitude física, como posição, velocidade, temperatura, luz, entre outras, em um sinal elétrico normalizado. Essa propriedade é utilizada principalmente por sensores. (TEIXEIRA, 2017).

Equação para determinação da distância

$$d = c.t / 2 \qquad (04)$$

Onde:

d = distância em metros entre o sensor e o objeto;

c = velocidade do som no ar em função da temperatura;

t = tempo do pulso ultrassônico até atingir o objeto e retornar ao sensor;

/ 2= divisão por 2, onda ultrassônica percorre o trajeto de ida e volta;

A variação da velocidade do som ***c*** em função da temperatura do ar, é calculada segundo a fórmula:

$$c = 331,45\sqrt{\frac{\vartheta}{273,15}} \qquad (05)$$

Onde:

331,45 = a velocidade do som a 0 graus Celsius (273,15 Kelvin);

ϑ = temperatura do ar (considerando-se o ar seco);

273,15 = temperatura kelvin (equivalente a 0 °C);

C = velocidade do som;

ρ = massa específica do ar;

Z = impedância acústica

Quadro 02: Influência da temperatura do ar na velocidade do som.

ϑ em °C (K)	c em m/s	C em km/h	ρ em kg/m³	Z em N·s/m³
-30 °C (243,15 K)	312,7	1.171,4	1,438	453,4
-25 °C (248,15 K)	315,9	1.171,4	1,413	449,1
-20 °C (253,15 K)	319,1	1.171,4	1,388	444,8
-15 °C (258,15 K)	322,2	1.171,4	1,363	440,6
-10 °C (263,15 K)	325,3	1.171,4	1,339	436,5
-5 °C (268,15 K)	328,4	1.182,6	1,316	432,4
0 °C (273,15 K)	331,5	1.193,4	1,293	428,3
5 °C (278,15 K)	334,5	1,204,2	1,269	424,5
10 °C (283,15 K)	337,5	1.215,0	1,247	420,7
15 °C (288,15 K)	340,5	1.226,0	1,225	417,0
20 °C (293,15 K)	343,4	1.237,0	1,204	413,5
25 °C (298,15 K)	346,3	1.246,7	1,184	410,0
30 °C (303,15 K)	349,2	1.257,12	1,164	406,6

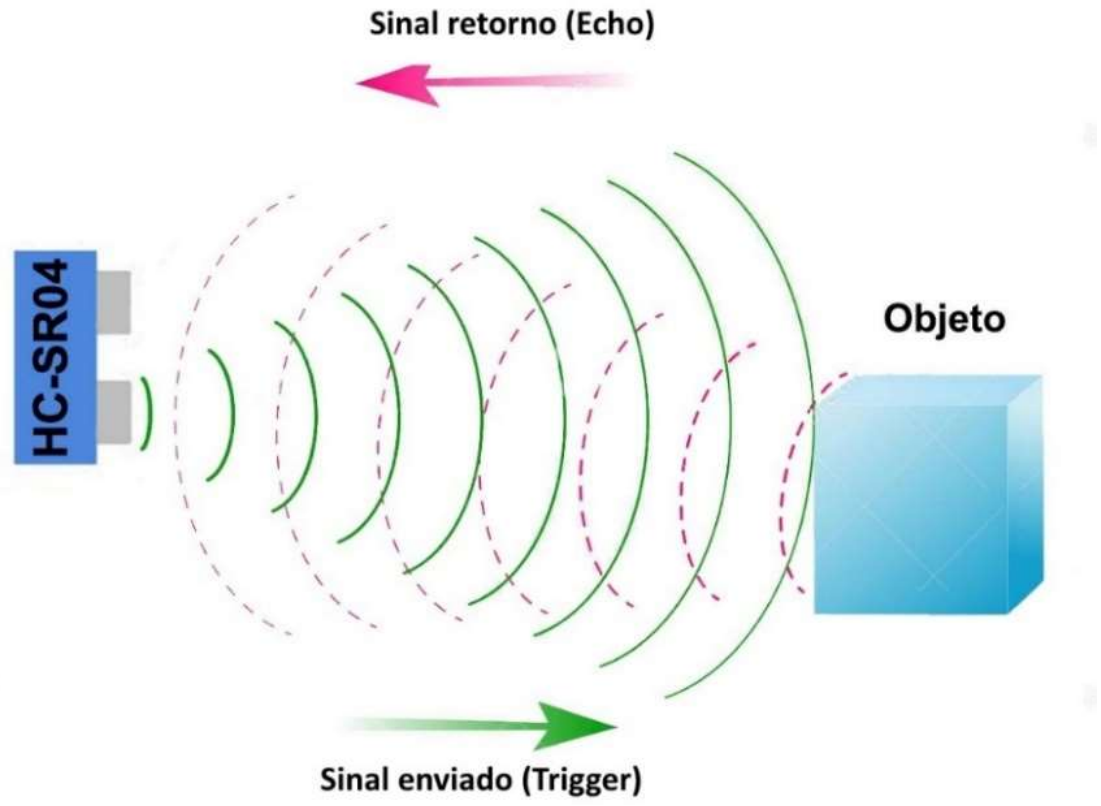
Fonte: Wikipedia

Em alguns países onde a temperatura é bem menor em relação ao nosso, a baixa temperatura em função da velocidade do som, pode ser afetada de forma a alterar o resultado da distância em metros do veículo em relação ao objeto, como mostra a tabela, e dependendo da precisão do sistema, se não se atentar para a temperatura, pode ser prejudicial. Mas neste projeto, será utilizada uma temperatura média para efetuar o cálculo da distância, onde a partir da temperatura de 10°C até 30°C (temperatura utilizada para média), obtemos uma temperatura média de 20°C, e se como exemplo utilizarmos um tempo de 10 ms e a temperatura média de 20°C com c=343,4 m/s como mostra a tabela, a distância será de 1,7m. Dessa forma desenvolveremos um sistema em linguagem C para calcular a distância entre o veículo ou objeto. (WIKIPEDIA, 2017).

3.11. Funcionamento

Segundo Datasheet (2011) o funcionamento do sensor HC-SR04 (Figura 20) se baseia no envio de sinais ultrassônicos pelo sensor, que aguarda o retorno (echo) do sinal, e com base no tempo entre envio e retorno, calcula a distância entre o sensor e o objeto detectado.

Figura 20: Envio e recepção de sinal do Sensor HC-SR04

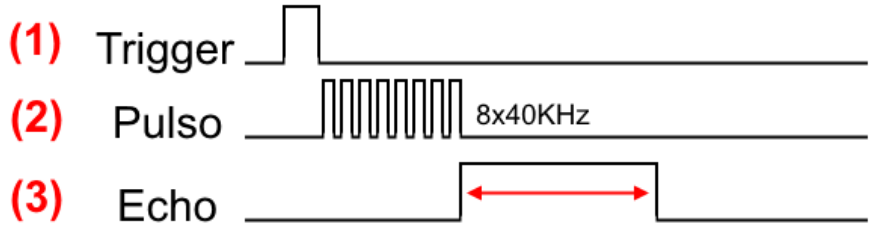


Fonte: Vidadesilicio

Primeiramente é enviado um pulso de 10µs (Figura 21), indicando o início da transmissão de dados. Depois disso, são enviados 8 pulsos de 40 KHz e o sensor então aguarda o retorno (em nível alto/high), para determinar a distância entre o sensor e o objeto, utilizando a equação:

Distância = (Tempo echo em nível alto * velocidade do som) / 2 (06)

Figura 21: Forma das ondas do sensor

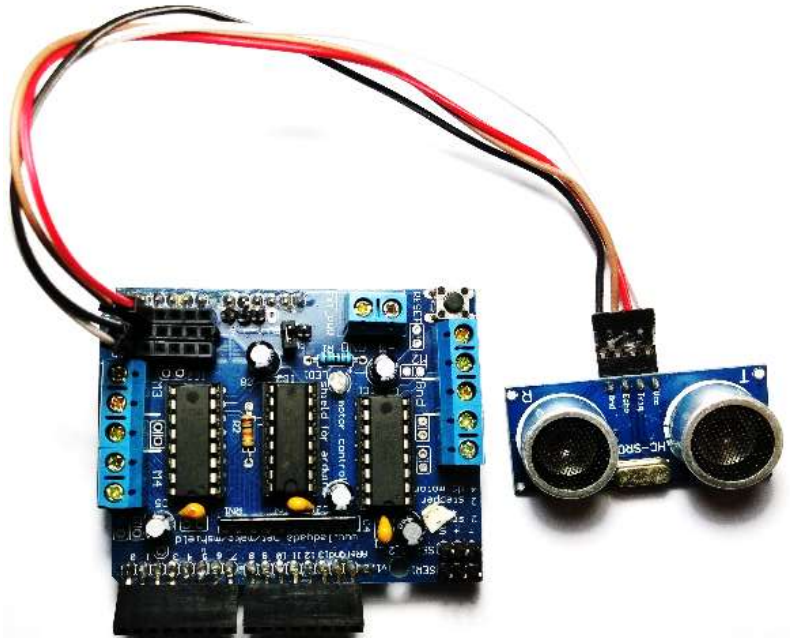


Fonte: Vidadesilicio

3.12. Conectando o Sensor HC-SR04 na placa Shield

A ligação do sensor HC-SR04 na placa Shield (figura 22) vai utilizar, a barra de pinos analógicos (figura 13): o fio vermelho conecta o pino A4 ao pino Trigger responsável pelo envio de sinais, fio marrom conecta o pino A5 ao pino Echo que recebe os sinais de volta, a alimentação será feita pelo pino VCC (5 V) de cor branco e o fio preto ligado ao GND.

Figura 22: Ligação do sensor HC-SR04 na placa Shield



Fonte: Autores

O programa usa a biblioteca Ultrasonic, que se encontra dentro da pasta LIBRARIES da IDE do Arduino.

3.13. SERVO MOTOR

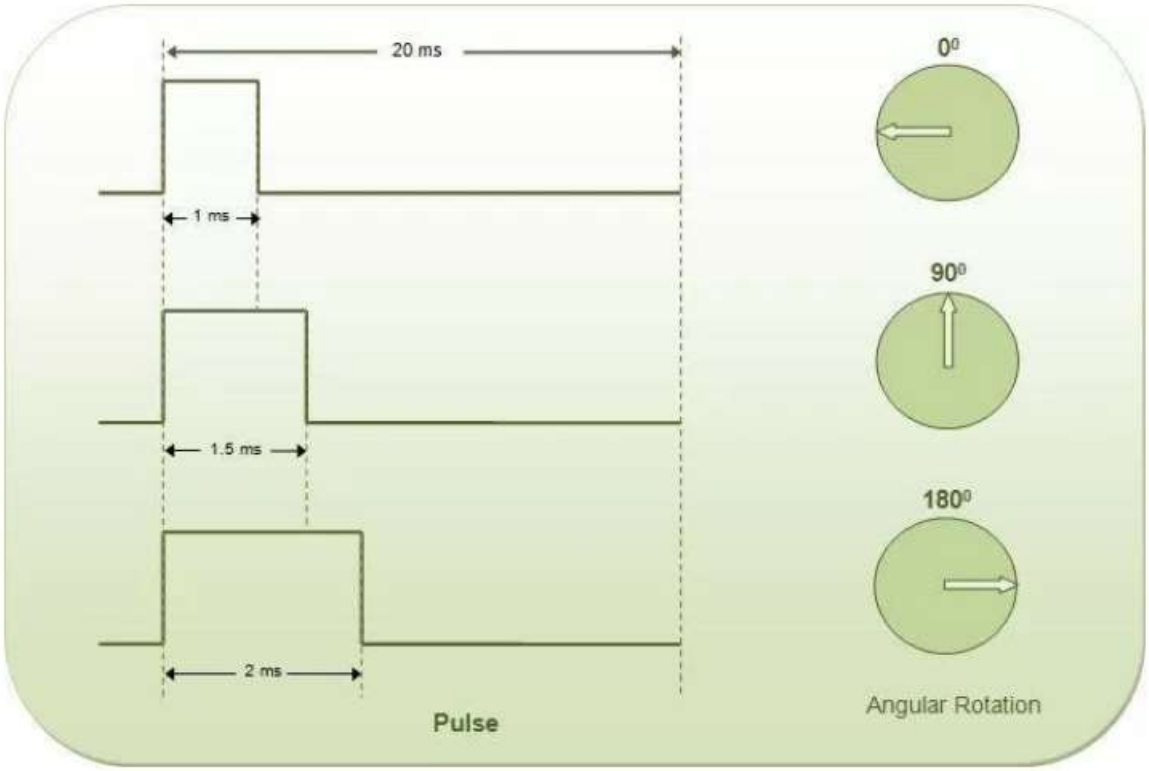
Vidadesilicio (2017). Entre os atuadores tem-se um motor bem especial. Os servo motores são muito utilizados quando o assunto é robótica. De forma simplificada, um servo motor é um motor na qual podemos controlar sua posição angular através de um sinal PWM. Dessa forma, um servo motor (figura 23) é um atuador eletromecânico utilizado para posicionar e manter um objeto em uma determinada posição. Para isso, ele conta com um circuito que verifica o sinal de entrada e compara com a posição atual do eixo.

Figura 23: Servo motor para Arduino

Fonte: Vidadesilicio

Diferentemente dos motores CC ou motores de passo que podem girar indefinidamente, os servo motores podem girar apenas 180º em torno de seu eixo como mostra a (figura 24).

Figura 24: Ângulos do servo motor



Fonte: Vidadesilicio

Servo motores geralmente possuem 3 pinos:

- Alimentação positiva (vermelho) – 5 V;
- Terra (Preto ou Marrom) – GND;
- (Amarelo, Laranja ou Branco) – Ligado a um pino digital de entrada e saída;

Segundo Vidadesilio (2017) servo motores consomem uma corrente significativa ao se movimentarem. A utilização de uma fonte externa pode ser necessária e é recomendada. Lembre-se de conectar o pino GND da fonte externa ao GND do Arduino para que a referência seja a mesma.

Apesar de sua posição ser controlada através do duty cycle de um sinal PWM enviado ao pino de controle não é necessária a conexão do pino de controle a um pino que possua PWM, pois será utilizada a biblioteca Servo.h.

A utilização de analog Write produzirá um controle de menor precisão e poderá até danificar alguns servos por sua frequência (490 Hz) ser 10 vezes superior a frequência típica de controle de alguns servos.

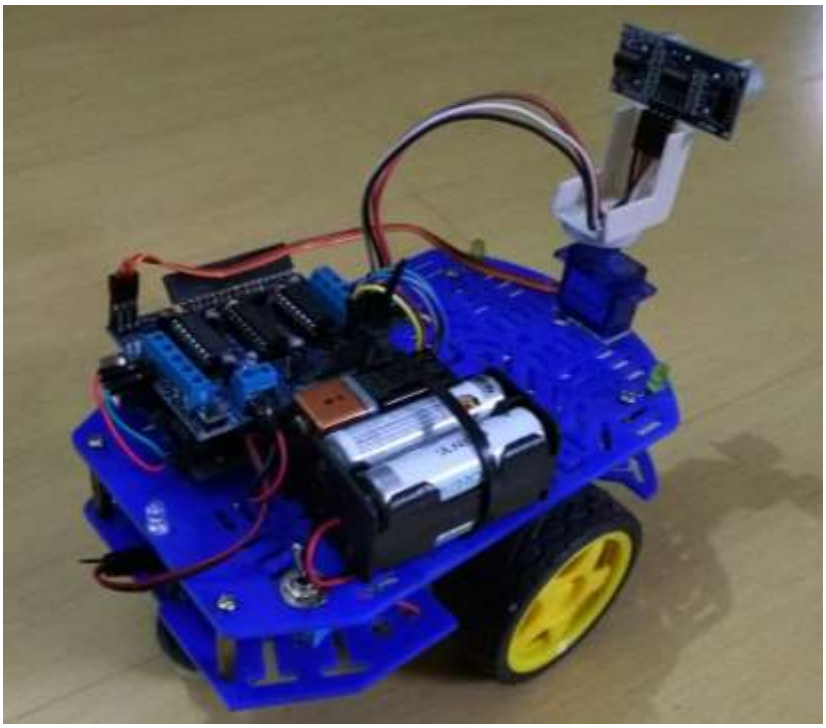
Além de mais preciso e recomendado, o uso da biblioteca Servo.h faz com que o uso do servo seja mais fácil. Isso se dá pelo fato de você só precisar definir o ângulo que você deseja, não necessitando o uso dos valores de PWM (0 a 255 bit)

4. Projeto

4.1. Princípio de funcionamento

O Carro Explorador Autônomo (figura 25) ao ser ligado, se move para frente a uma velocidade determinada pelo programador, dando início ao programa (figura 26).

Figura 25: Carro explorador aautônomo



Fonte: Autores

O veículo conta com um sensor ultrassônico HC-SR04 posicionado na parte frontal do veículo e montado sobre um servo motor, responsável pela movimentação do sensor para ambos os lados do veículo (direito e esquerdo), este sensor vai monitorar a distância a frente do nosso veículo e ao se aproximar de um obstáculo a uma distância pré determinada por uma programação, o carro para de se mover.

A Central recebe os sinais e envia um comando para o servo motor localizado na parte frontal do veículo para movimentar o sensor, primeiramente para o lado esquerdo, faz a leitura da distância e armazena seu valor, em seguida movimenta o

sensor para o lado direito, faz a leitura e armazena seu valor e envia para a Central poder comparar os valores obtidos e após calcular qual distância é maior, o CEA vai tomar a decisão de seguir o caminho onde a distância é maior com o propósito de encontrar mais segurança para continuar seu trajeto. Após a tomada de decisão, o CEA recua por um pequeno espaço acendendo um led branco localizado em sua traseira simulando um veículo dando marcha a ré, em seguida irá acender um dos led's amarelos localizados nas laterais do veículo informando assim para qual dos lados o CEA vai se direcionar, para direita ou para esquerda. A partir desse momento o ciclo é novamente reiniciado.

4.2. Problemas e instabilidades

Durante a construção do Carro Explorador Autônomo encontrou-se alguns problemas que tiveram que ser corrigidos:

- Correção de velocidades das rodas, a fim de manter o carro em linha reta.
- Correção da tensão de alimentação da placa que estando abaixo da ideal (7 V), provoca instabilidades e mau funcionamento.
- Correção das conexões e encaixes para que não atrapalhe o funcionamento de sensores e atuadores (HC-SR04, Servo e Motores DC).
- Devido ao ângulo de efeito do sensor ultrassônico de aproximadamente 15º, há a possibilidade de não captação de obstáculos que estiverem abaixo do nível do sensor.
- Obstáculos com inclinações elevadas também podem não serem detectados a uma determinada distância pela possibilidade de o receptor não receber a onda de volta. Obstáculos com menos de 0,5 m² podem não serem detectados por este sensor.

5. Conclusão

Este trabalho teve como meta desenvolver um carro protótipo que estivesse o mais próximo possível da realidade, foi utilizada a plataforma Arduino para implementação do Carro Explorador Autônomo devido:

Pela diversidade de material para construção de um veículo em escala reduzida; possui um software próprio e nos permite fazer modificação e implementações; materiais encontrados em diversos sites e lojas de equipamentos eletrônicos; esse tipo de plataforma tem seu uso liberado totalmente gratuito sem necessidade de comprar uma licença.

Durante a montagem do nosso protótipo, ocorreu alguns erros no processo de programação para execução dos comandos necessários, teve dificuldade no envio de comandos para os atuadores, as rodas não tinha o mesmo sincronismo, o carro não segue em linha reta, não faz a parada com exatidão e não reconhece alguns dos obstáculos. Depois de muita pesquisa e executar vários testes, conseguimos realizar os ajustes e correções na programação com relação a diferença de rotação em uma das rodas o que permite o carro sair fora da linha reta, ocasionado pela diferença na voltagem que a placa Arduino envia para os dois motores acoplados as rodas, alterou-se a velocidade e tempo de resposta de uma das rodas, junto a sua programação e teve o sincronismo recuperado, foi alterado também a posição do sensor, elevou-se a sua altura em relação ao solo para poder ampliar seu campo de visão e assim reconhecer um obstáculo com mais precisão e poder parar com segurança, foi acrescentado um led branco para sinalizar quando o carro estiver recuando, juntamente com outros dois led’s amarelos, um do lado direito e o outro do lado esquerdo para que informe sobre a mudança de direção após ser tomada uma decisão.

Após todas as correções e ajustes feitos, os objetivos em relação ao Carro Explorador Autônomo foram atingidos.

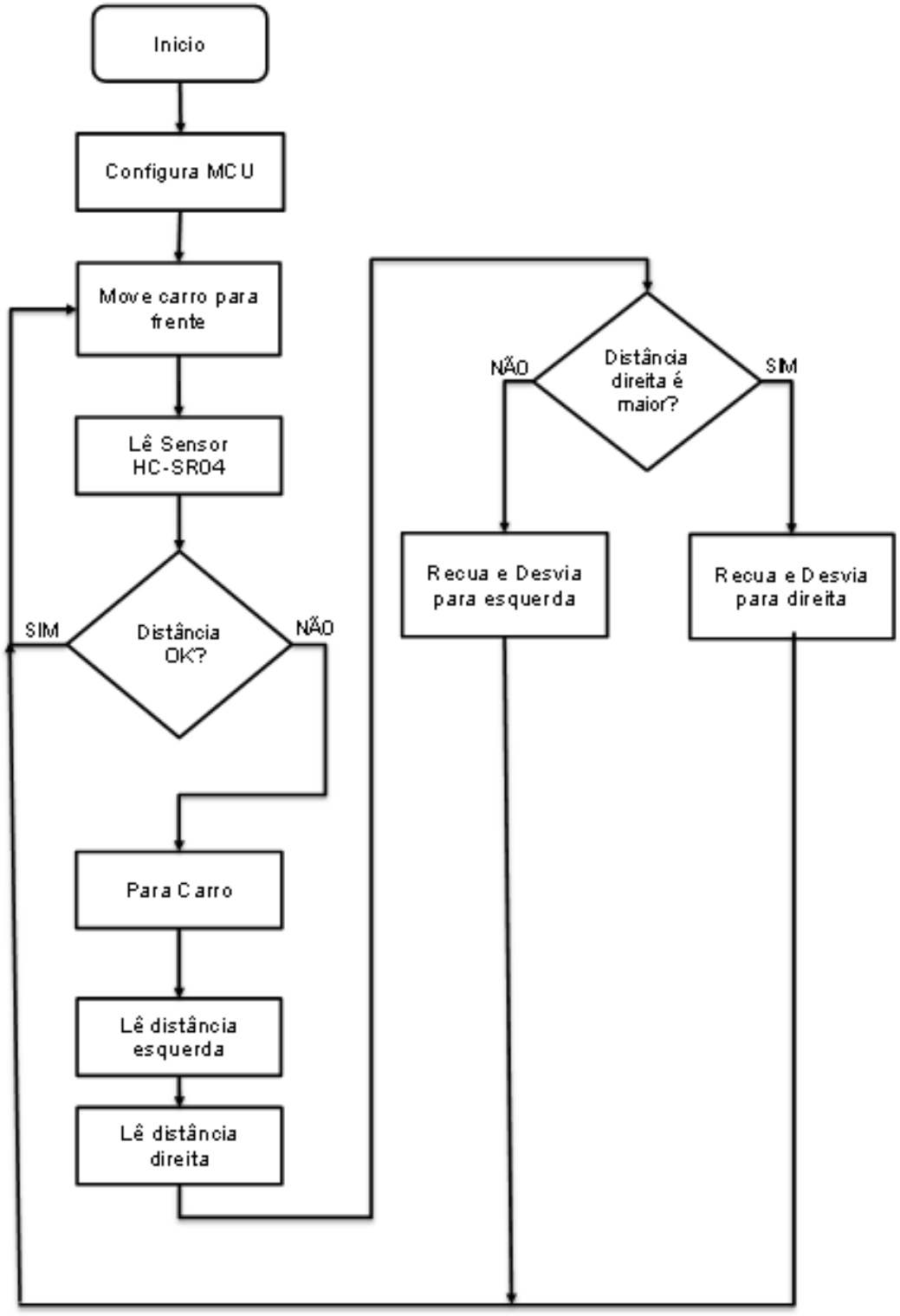
5.1. Trabalhos futuros

- Implementação da programação para que o carro possa desviar e seguir frente sem necessidade de parar;
- Instalar mais sensores para se ter maior controle externo do ambiente;
- Adaptar uma direção elétrica para acionamento de ambas as rodas dianteiras;
- Construir um modelo em escala maior;
- Utilizar recursos de GPS.

6. Apêndice

6.1. Fluxograma de funcionamento

Fluxograma 01: Funcionamento do Carro Explorador Autônomo



Fonte: Autores

6.2. Código Fonte

```
                        // --- Bibliotecas Auxiliares ---
#include <AFMotor.h>                                    //Inclui biblioteca AF Motor
#include <Servo.h>     //Inclui biblioteca para controle de Servos

                        // --- Seleção dos Motores ---
AF_DCMotor motor1(1);                                   //Seleção do Motor 1
AF_DCMotor motor3(3);                                   //Seleção do Motor 3

                        // --- Mapeamento de Hardware ---
#define serv 10                                         //controle do Servo 1
#define trig A4                                 //Saída para o pino de trigger do sensor
#define echo A5                                  //Entrada para o pino de echo do sensor
#define ledBr A3                                        //Saída para o led branco
#define ledAm A2                                 //Saída para o led amarelo lado direito
#define ledAmm A1                               //Saída para o led amarelo lado esquerdo

          // --- Protótipo das Funções Auxiliares ---
float medirDistancia();         //Função para medir, calcular e retornar a distância em cm
void triggerPulso();                        //Função que gera o pulso de trigger de 10µs
void decide();                     //Função para tomada de decisão. Qual melhor caminho?
void carro_frente();                         //Função para movimentar carro para frente
void carro_re();                             //Função para movimentar carro para trás
void carro_esquerda();                      //Função para movimentar carro para esquerda
void carro_direita();                        //Função para movimentar carro para direita
void carro_para();                                  //Função para parar o carro
void pisca_ledBr();                                 //Função para piscar led branco
void pisca_ledAm();                        //Função para piscar led amarelo lado direito
void pisca_ledAmm();                      //Função para piscar led amarelo lado esquerdo
```

```
                                    // --- Objetos ---
Servo servo1;                                          //Cria objeto para o servo motor

                                  // --- Variáveis Globais ---
float dist_cm;            //Armazena a distância em centímetros entre o robô e o obstáculo
float dist_direita;                     //Armazena a distância em centímetros da direita
float dist_esquerda;                     //Armazena distância em centímetros da esquerda

                                // --- Configurações Iniciais ---
void setup()
{
  //A bibl conf as entradas e saídas pertinentes ao Motor Shield...
 pinMode(trig, OUTPUT);                                    //Saída para o pulso de trigger
 pinMode(serv, OUTPUT);                                        //Saída para o servo motor
 pinMode(echo, INPUT);                                    //Entrada para o pulso de echo
 pinMode(A3,   OUTPUT);                                           //Saída para led branco
 pinMode(A2,   OUTPUT);                              //Saída para led amarelo lado direito
 pinMode(A1,   OUTPUT);                             //Saída para led amarelo lado esquerdo

 servo1.attach(serv);                              //Objeto servo1 no pino de saída do servo
 digitalWrite(trig, LOW);                                  //Pino de trigger inicia em low
 servo1.write(80);                                                  //Centraliza servo
 delay(500);                                     //Aguarda meio segundo antes de iniciar
}

                                    // --- Loop Infinito ---
void loop()
{
     carro_frente();
```

```
    delay(80);
    dist_cm =  medirDistancia();
    if(dist_cm < 20) //distância menor que 20 cm?
    {
        decide();
    }
}

                    // --- Desenvolvimento das Funções Auxiliares ---

float  medirDistancia()                    //Função que retorna a distância em centímetros
{
  float pulso;        //Armazena o valor de tempo em µs que o pino echo fica em nível alto
  triggerPulso();                        //Envia pulso de 10µs para o pino trigger do sensor
  pulso = pulseIn(echo, HIGH);    //Mede o tempo em que echo fica em nível alto e armazena
na variável pulso
  /*
                          >>> Cálculo da Conversão de µs para cm:
   Velocidade do som = 340 m/s = 34000 cm/s
   1 segundo = 1000000 micro segundos

      1000000 µs - 34000 cm/s
            X µs - 1 cm
            X = 1E6 / 34000 = 29.41

Para compensar o ECHO (ida e volta do ultrassom) multiplica-se por 2

    X' = 29.41 x 2 = 58.82
 */

  return (pulso/58.82);                    //Calcula distância em cm e retorna o valor
```

```
}

void triggerPulso()           //Função para gerar o pulso de trigger para o sensor HC-SR04
{
   digitalWrite(trig,HIGH);                              //Saída de trigger em nível alto
   delayMicroseconds(10);                                                  //Por 10µs ...
   digitalWrite(trig,LOW);                            //Saída de trigger volta a nível baixo
}

void decide()   //Compara as distâncias e decide qual melhor caminho a seguir
{
   carro_para();                                                            //Para o carro
   delay(300);                                                            //Aguarda 300ms
   servo1.write(0);                             //Move sensor para direita através do servo
   delay(300);                                                            //Aguarda 300ms
   dist_direita = medirDistancia();             //Mede distância e armazena em dist_direita
   delay(1500);                                                          //Aguarda 1500ms
   servo1.write(175);                          //Move sensor para esquerda através do servo
   delay(300);                                                            //Aguarda 300ms
   dist_esquerda = medirDistancia();          //Mede distância e armazena em dist_esquerda
   delay(1500);                                                          //Aguarda 1500ms
   servo1.write(80);                                                    //Centraliza servo
   delay(300);
   if(dist_direita > dist_esquerda)          //Distância da direita maior que da esquerda?
   {                                                                              //Sim...
      pisca_ledBr();                                                   //Pisca led branco
      delay(200);                                                             //Por 200ms
      carro_re();                                                  //Move o carro para trás
      delay(200);                                                             //Por 500ms
      pisca_ledAm();                                        //Pisca led amarelo lado direito
```

```
      carro_direita();                                          //Move o carro para direita
      delay(500);            //Por 500ms ...Valor pode alterar em relação a carga bateria
      carro_frente();                                           //Move o carro para frente
   }
   else                                                                          //Não...
   {
      pisca_ledBr();                                                  //Pisca led branco
      delay(200);                                                             //Por 200ms
      carro_re();                                                 //Move o carro para trás
      delay(200);                                                             //Por 200ms
      pisca_ledAmm();                                       //Pisca led amarelo lado esquerdo
      carro_esquerda();                                     //Move o carro para esquerda
      delay(500);                                                             //Por 500ms
      carro_frente();                                           //Move o carro para frente
   }

}
void carro_frente()
{
     motor1.setSpeed(150);
     motor1.run(FORWARD);
     motor3.setSpeed(200);
     motor3.run(FORWARD);

}
void carro_re()
{
     motor1.setSpeed(150);
     motor1.run(BACKWARD);
     motor3.setSpeed(200);
```

```
      motor3.run(BACKWARD);

}
void carro_esquerda()
{
      motor1.setSpeed(150);
      motor1.run(FORWARD);
      motor3.setSpeed(200);
      motor3.run(BACKWARD);
}
void carro_direita()
{
      motor1.setSpeed(150);
      motor1.run(BACKWARD);
      motor3.setSpeed(200);
      motor3.run(FORWARD);
}
void carro_para()
{
      motor1.setSpeed(0);
      motor1.run(RELEASE);
      motor3.setSpeed(0);
      motor3.run(RELEASE);

}
void pisca_ledBri()
{
  digitialWrite(A3,HIGH);
  delay(200);
  digitalWrite(A3,LOW);
```

```
  delay(200);
  digitalWrite(A3,HIGH);
  delay(200);
  digitalWrite(A3,LOW);
  delay(200);
}
void pisca_ledAm()
{
  digitalWrite(A2,HIGH);
  delay(200);
  digitalWrite(A2,LOW);
  delay(200);
  digitalWrite(A2,HIGH);
  delay(200);
  digitalWrite(A2,LOW);
  delay(200);
}
void pisca_ledAmm()
{
  digitalWrite(A1,HIGH);
  delay(200);
  digitalWrite(A1,LOW);
  delay(200);
  digitalWrite(A1,HIGH);
  delay(200);
  digitalWrite(A1,LOW);
  delay(200);
}
```

7. Referências

Arduino & Cia – Arduino Uno
http://www.arduinoecia.com.br
acesso em: 08/2017

BASTOS, J. P. A. Eletromagnetismo para engenharia: estática e quase estática. Edição 3. Florianópolis: Editora da UFSC, 2004. 398 p - https://repositorio.ufsc.br/xmlui/bitstream - acesso em: 09/2017

Bayard, Oswaldo - Carro Autônomo: Entendendo Essa Tecnologia
http://mais.uol.com.br/view/16279346 - acesso em: 09/2017

Cardoso, Daniel – Aprenda a controlar a velocidade de um motor DC
www.vidadesilicio.com.br - acesso em: 10/2017

CORRÊA, Marco Aurélio.Servoacionamentos e servomotores. Disponível em: < http://www.mecatronicaatual.com.br/secoes/leitura/494> Acesso em: 09/2017

Datasheet – Atmega 328 – acesso em: 09/2017
http://www.alldatasheet.com/view.jsp?Searchword=Atmega328%20datasheet

Futurecom- Inspiring Innovation- acesso em: 09/2017
http://www.futurecom.com.br/pt/sitemap.html

Hoepers, Rodrigo – Veiculo autônomo pdf – Universidade do Vale do Itajai
http://siaibib01.univali.br/pdf/Rodrigo%20Hoepers.pdf – acesso 09/2017

Kalatec Automação - http://www.kalatec.com.br/o-que-sao-motores-dc/ - acesso em 09/2017

Macedo, Felipe - material e vídeos
https://www.filipeflop.com - acesso em: 09/2017

Mota, Allan Deangelle. Apostila Arduino Básico: Vol. 1. Serra – ES: Vida de Silício, 2015. 40p. - www.vidadesilicio.com.br - acesso em: 09/2017

NET Computadores - https://netcomputadores.com.br/p/aunor3-arduino-uno-r3-/14715 - acesso em: 09/2017

Observatório Vaz Tolentino - http://vaztolentino.com.br/layout/vaztolen/logo-site.png
acesso em: 10/2017

Siemens LTDA - Unidade Automação e Controle – Acionamentos e Motores Elétricos Edição 01.2006 - www.siemens.com.br/motores - acesso em: 08/2017

Siemens – Motores de corrente continua
www.mariloureiro.net/tecnica/electrif/Motores_CC
acesso em: 09/2017

TEIXEIRA, Mariane Mendes. "Transdutor"; *Brasil Escola*. Disponível em <http://brasilescola.uol.com.br/fisica/transdutor.htm>. 08/2017

THE SPEED OF SOUND IN OTHER MATERIALS. NDT Resource Center.
https://pt.wikipedia.org/wiki/Velocidade_do_som - acesso em: 09/2017

Thomsen, Adilson - Como utilizar o sensor ultrasônico HC-SR04 -
http://buildbot.com.br/blog/como-utilizar-o-sensor-ultrasonico-hc-sr04/
acesso em: 10/2017

Udacity Brasil – Como funciona carro autonomo -
https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwiCnpT9i7XXAhVDTJAKHbawBNIQFggnMAA&url=https%3A%2F%2Fbr.udacity.com%2Fblog%2Fcomo-funciona-carro- acesso em: 08/2017

Vida de Silício Tutoriais - http://www.vidadesilicio.com.br/ - acesso em: 10/2017

WEBSTER, JOHN G. Measurements, Instrumentation, and Sensors. Boca Raton: Crc Press Llc, 1999. http://www.kelm.ftn.uns.ac.rs/literatura/si/pdf - acesso em: 08/2017