FUNDAÇÃO OSWALDO ARANHA
CENTRO UNIVERSITÁRIO DE VOLTA REDONDA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA
TRABALHO DE CONCLUSÃO DE CURSO

IURY REIS E VILAÇA
LUCAS DA SILVA ARAÚJO

AUTOMATIZAÇÃO DE VEÍCULOS DE CARGAS INDUSTRIAIS GUIADOS VIA MÓDULO GPS

VOLTA REDONDA
2020

FUNDAÇÃO OSWALDO ARANHA
CENTRO UNIVERSITÁRIO DE VOLTA REDONDA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA
TRABALHO DE CONCLUSÃO DE CURSO

AUTOMATIZAÇÃO DE VEÍCULOS DE CARGAS INDUSTRIAIS GUIADOS VIA MÓDULO GPS

Monografia apresentada para o Curso de Engenharia Elétrica do UniFOA como requisito à obtenção do título de bacharel em Engenheiros Eletricistas

Alunos:
Iury Reis e Vilaça
Lucas da Silva Araújo

Orientador:
Prof. Aloano Régio De Almeida Pereira

VOLTA REDONDA
2020

FOLHA DE APROVAÇÃO

Alunos:

Iury Reis e Vilaça

Lucas da Silva Araújo

Título da Monografia: Veículo autônomo guiado via módulo GPS

Orientador: Prof. Aloano Régio De Almeida Pereira

Banca Examinadora

_____

Prof. Esp. Aloano Régio De Almeida Pereira

_____

Prof. Msc. Claúdio Marcio de Freitas da Silva

_____

Prof. Msc. Edson de Paula Carvalho

A Deus, nossos pais e amigos que de certa forma contribuíram para alcançar nossos objetivos.

AGRADECIMENTOS

A Deus, por nos ter guiado e dado sabedoria e saúde até aqui para concluir mais esse desafio.

Aos nossos pais por todo cuidado, educação que nos foi dada e se estamos aqui completando mais essa etapa, devemos muito a eles.

A todos os professores pelo conhecimento passado com seriedade e dedicação, em especial nosso professor orientador Prof. Aloano Régio De Almeida Pereira que se empenhou ao máximo em nos direcionar pelo melhor caminho.

RESUMO

O crescente desenvolvimento e revolução das indústrias tornaram o mercado extremamente competitivo, com isso o setor industrial tem se modernizado criando tecnologias, ideias e buscando sempre a otimização do processo de produção tornando-o otimizado, seguro, minimizando os erros.

Com a abertura econômica de países, até então fechados nos últimos anos, com aumento brusco da competitividade, com foco na redução de desperdícios e o advento da automação juntamente revolução industrial 4.0, com o avanço da tecnologia, as engenharias de todas as áreas e a ciência vem desempenhando um papel fundamental na inovação e na otimização da linha de produção. Essa sede de inovação, revolucionou o mercado tornando oportuno o surgimento de novos métodos de produção cada vez mais eficientes, limpos e em conformidades com leis ambientais, dando continuidade a essa ideologia o destaque fica por conta da aplicação do sistema de veículo guiado de forma autônoma, utilizado principalmente para transporte de materiais.

O estudo tem o objetivo de desenvolver um veículo guiado automaticamente por GPS com alarme de falha, sensores ultrassônicos de aproximação de obstáculo. O veículo terá as coordenadas pré-programadas fazendo que sua rota seja estabelecida de forma que ele percorra todo seu trajeto sem interrupção e sem a necessidade cabos ou fitas magnéticas para guiá-lo de forma segura.

Nesse projeto o Arduino receberá os sinais de tensão, ora pelo módulo GPS ora pelo sensor ultrassônico. Esses sinais serão capitados pelas entradas, que posteriormente serão processadas pela programação existente em sua memória realizando a leitura da sua posição ou se possuem algum obstáculo a frente para dar início a rota pré-determinada.

Embora, o projeto possuir uma grande aplicabilidade, o seu foco será guiar veículos de forma autônoma utilizando coordenadas geográficas visando eficiência na movimentação de carga sem sofrer interferências e capacidade de adaptação em qualquer planta.

Palavras-chave: Veículo autônomo Guiado, AGV, Arduíno, GPS.

ABSTRACT

The growing development and revolution of the industries becoming an extremely competitive market, with this the industrial sector has been modernizing creating new technologies, ideas and always seeking the optimization of the production process making it optimized, safe, minimizing errors.

With the economic opening of countries, hitherto closed in recent years, with a sharp increase in the previous one, with focus on reducing waste and the advent of automation along with the industrial revolution 4.0, with the advancement of technology, such as engineering in all areas and science has been playing a fundamental role in innovation and in the optimization of the production line. This thirst for innovation, revolutionized the market making it opportune for the emergence of new production methods that are increasingly more efficient, clean and in compliance with environmental laws, continuing this ideology, the highlight is the application of the autonomously guided vehicle system, mainly used for material transportation.

The study aims to develop a vehicle automatically guided by GPS with failure alarm, ultrasonic sensors for approaching obstacles. The vehicle will have pre-programmed coordinates making its route chosen so that it travels along its entire route without interruption and without the need for cables or magnetic tapes to guide it safely.

In this project, the Arduino will receive the voltage signals, either by the GPS module or by the ultrasonic sensor. These signals will be captured by the inputs, which will later be processed by the existing programming in its memory, reading the position or if there are any obstacles ahead to start the predetermined route.

Although, the project has a great applicability, its focus will be to guide vehicles autonomously using geographic coordinates engineering in cargo handling without suffering interference and adaptability in any plant.

Key words: Automated Guided Vehicle, AGV, Arduino, GPS.

LISTAS DE ABREVIATURAS E SIGLAS

A - Ampere

Ah - Ampere Hora

ANATEL - Agência Nacional de Telecomunicações

BI - Business Intelligence (Inteligência das Coisas)

BSC - Base Controller Station (Estação de Controle Base)

BSS - Base Station System (Sistema de Estação Base)

DHCP - Dynamic Host Configuration Protocol (Protocolo de Configuração Dinâmica de Host)

EEPROM - Electrically Eraseble Programmable Read Only Memory (Memória de Leitura Eletricamente Apagável e programável)

ECEF – Earth-Centered Earth-Fixed Coordinate System (sistema de coordenadas geográficas e cartesianas)

ERP - Enterprise Resource Planning (sistemas de controle administrativo)

FM - Frequency Modulation (Frequência Modulada)

GHz - Giga Hertz

GPS - Global Position System (Sistema de Posicionamento Global)

IDE - Integrated Development Environment (Ambiente de Desenvolvimento Integrado)

IP - Internet Protocol (Protocolo de Internet)

IPEA - Instituto de Pesquisa Econômica Aplicada

KHz - Kilo Hertz

LED - Light Emitting Diode (Diodo Emissor de Luz)

Li - Lítio

MHz - Mega-Hertz

MS - Mobile Station (Estação Móvel)

NiCd - Níquel Cádmio

NMEA - National Marine Electronics Association (Associação Nacional de Eletrônica Marinha)

NSS - Network Switching System (Sistema de Comutação de Rede)

PWM - Pulse Widht Modulation

RF - Rádio Frequência

TTL – Transistor-Transistor-Logic (Lógica Transistor-Transitor)

UHF - Ultra High Frequency (Frequência Ultra Alta)

LISTAS DE FIGURAS

LISTAS DE FIGURAS

LISTAS DE TABELAS

ANEXOS

1 INTRODUÇÃO

1.1 Tema

Veículo Guiado Automaticamente ou AGV (do inglês, *Automated Guided Vehicle*) é um robô industrial móvel usado para o manuseio, manipulação de equipamentos ou materiais exercendo a função de transportadores, elevadores e guindastes entre outros. São construídos para realizar tarefas específicas de transferência de carga e/ou peças de uma linha de produção e são capazes de mover ao longo de um determinado caminho deixando o processo otimizado, algumas vezes são desenvolvidos para executar tarefas impossíveis de serem executadas de forma segura por humanos.

No início da automação industrial os processos eram constituídos por pequenas ilhas automatizadas, onde a intervenção e operação humana era indispensável para sincronizar todas as operações fazendo o processo ser unificado.

Após a década de 1960, com o início do desenvolvimento e crescente evolução dos equipamentos com unidades de processamento e instrumentos com comunicação em tempo real, tornou-se possível a automatização de sistemas melhorando a otimização dos sistemas de movimentação de materiais, que são estão diretamente relacionados com o aumento da competitividade, sendo uma das funções de maior relevância na logística de uma empresa e por esse motivo torná-lo eficiente tornou-se fundamental.

1.2 Justificativa

A iniciativa desse projeto de pesquisa tem o propósito de atuar nos setores indústriais seja ele de pequeno, médio ou grande porte, visando a otimização e automatização da movimentação de cargas e materiais sem a intervenção humana.

Nos últimos anos com o avanço tecnológico e uso de robótica, o transporte de materiais tem se tornado cada vez mais eficientes e são projetados conforme as necessidades e características específicas da planta.

Nesse trabalho apresentamos um projeto de AGV guiado por coordenadas geográficas sendo completamente adaptável em grande parte dos setores da indústria.

Desta forma, esse trabalho visa estudar o desenvolvimento de comando e controle de AGV's guiados por coordenadas geográficas, sendo possível a implementação em qualquer veículo usado no transporte de cargas alterando sua configuração de condução do AGV com outra tecnologia, apresentando suas vantagens, eficiência na movimentação de carga sem sofrer interferências e capacidade de adaptação em qualquer planta.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo geral é estudar as tecnologias de desenvolvimento utilizando plataformas eletrônicas para a construção de um Veículo Auto Guiado (AGV), convergindo para o desenvolvimento de um protótipo de AGV capaz de se deslocar automaticamente seguindo uma trajetória pré determinada.

1.3.2 Objetivo Específico

O objetivo específico deste projeto consiste no desenvolvimento de um protótipo de controle e comando de veículo guiado automaticamente capaz de percorrer uma rota utilizando coordenadas geográficas via modulo GPS, incluindo sensores de aproximação, sinais de alerta e leds de sinalização. Serão utilizados conceitos de robótica com implementação de baixo custo para desenvolvimento do *hardware* e *software*. Este objetivo surge da necessidade de reduzir o congestionamento provocado pelo fluxo de materiais, que afeta o fluxo de matéria-prima que abastece as áreas de produção oferecendo segurança e velocidade em operações ininterruptas, além de favorecer as condições acústicas e ambientais, ou seja, sem ruído e sem poluição.

1.4 Metodologia

Com o intuito de concluir esse trabalho foram realizados os respectivos tópicos a seguir:

- Pesquisa bibliográfica a respeito de veículos autoguiados, tanto com respeito a forma de construção, bem como as técnicas de controle dos mesmos.
- Estudo do funcionamento e programação da placa controladora Arduino.
- Levantamento das necessidades do setor industrial e modo de implementação.
- Estudo de caso apresentando os benefícios que esta tecnologia proporciona ao processo de logística e otimização de produção de uma empresa.

2 TRANSPORTE E MOVIMENTAÇÃO DE CARGAS

2.1 Automação industrial

Segundo Muraro (1969) desde os princípios da humanidade, o homem conserva uma vontade gigantesca de se desenvolver, em uma busca infinita por facilidades que tornem a sua vida mais simples e segura.

A partir dessa necessidade é possível observar que a revolução industrial foi um fator natural desse desenvolvimento. Esta foi a transição para novos processos de manufatura. Esta transformação incluiu a transição de métodos de produção artesanais para a produção por máquinas, motivados pela expansão dos mercados que se tornaram globais e não mais locais. Com o advento da primeira revolução industrial, marco na evolução mundial, surge à possibilidade de expandir a produção e, assim, confeccionar objetos de maior qualidade a preços reduzidos. Para que tais escalas de produção fossem atingidas o uso somente de força braçal não se fazia suficiente, nascia assim, um processo denominado mecanização das etapas de produção, com o objetivo de um aumento significativo na produtividade da indústria. Assim, a mecanização é uma técnica que permite a extensão das funções humanas – que não a cerebral – através de um processo de fragmentação e tem como principal característica a possibilidade de repetição infinita de determinados movimentos (MURARO, 1969).

2.2 A indústria 4.0 e a implementação de AGV’s

O termo "Indústria 4.0" tornou-se conhecido em 2011, quando foi criada uma iniciativa denominada "Industrie 4.0" - uma associação de representantes do mundo empresarial, político e acadêmico que apoiou a ideia como uma abordagem para fortalecer a competitividade da indústria manufatureira alemã.

Um dos principais objetivos centrais da indústria 4.0 é tornar a produção mais eficaz otimizando processo, reduzindo custos, menos tempos de equipamentos ociosos por manutenção e setups de linha de produção, tornando os resultados mais competitivos.

Podem ser utilizadas novas tecnologias para integrar homem e máquina, onde o homem para de realizar atividades pesadas e repetitivas para se preocupar com o

planejamento da produção, com isso, as máquinas são capazes de utilizar várias informações geradas pela própria planta (*machine to machine*) para se realinhar e se tornando resiliente a falhas e falta de suprimento e outras interrupções que atualmente exigem a intervenção dos humanos.

O uso de dispositivos de Internet das Coisas (*Internet of Things*), Big Data e virtualização fazem com que os equipamentos se tornem mais “inteligentes” fornecendo e recebendo informações dos sistemas de controle administrativo (ERP - Enterprise Resource Planning) e Inteligência do Negócio (BI - Business Intelligence) para se modelar as necessidades do cliente e ao mesmo tempo providenciando uma produção mais enxuta.

2.3 Sistemas de Localização Por Satélite

O princípio da navegação por satélites do sistema GPS é baseado no conceito de análise do tempo de chegada de um sinal (*TOA, time of arrival*) (Kaplan, 1996). O conceito de TOA constitui em transmitir um sinal em um determinado tempo conhecido e medir o tempo correspondente a chegada deste sinal em um tempo posterior, predeterminado. Então, este período de tempo é multiplicado pela velocidade de propagação do sinal para se ter a distância entre o emissor e o receptor. Se o relógio do receptor estiver sincronizado com os relógios dos satélites GPS, em posições definidas e com as medidas de três diferentes satélites GPS, é concedido ao usuário calcular sua posição.

3 VEÍCULO GUIADO AUTOMATICAMENTE (AGV)

3.1 Definição de AGV (*Automated Guided Vehicle*)

O *Automated Guide Vehicle* ou veículo guiado automaticamente, está presente há algum tempo na indústria em diversas áreas de atuação, entretanto vem ganhando maior destaque com o avanço da tecnologia, integrando sensores, navegação por indução e de certa forma interagindo diretamente com processo sendo um equipamento acessível as empresas, seja ela de pequeno ou grande porte.

Conceitualmente, sua função é transportar materiais, equipamentos, peças de automóveis de forma autônoma, ou seja, sem interação humana. O piloto é substituído por uma software capaz de realizar uma rota determinada fazendo carga e descarga de forma autônoma. O transporte pode ser pequenos equipamentos ou ferramentas ou até grandes cargas como chassis de automóveis de grande porte.

Com a implementação da revolução industrial 4.0, o AGV torna-se um aliado poderoso para otimizar o processo produtivo, aliando precisão, automatização, velocidade de produção e conectividade com a produção com todo o processo.

3.2 Tipos de AGV

No mercado atual já existem diversos modelos de AGV e diferentes tecnologias aplicadas em cada um deles, entretanto, é necessário conhecer o processo ou a linha de produção ao qual será implementado para dimensioná-lo de forma segura, econômica, levando sempre em consideração suas vantagens e desvantagens incluindo suas limitações, seja ele guiado por trilhos, faixa magnética ou qualquer outra tecnologia. Diante disso, os AVG são desenvolvidos de acordo com a necessidade do processo, sendo assim, cada um possui características que são usadas como determinantes na escolha de qual tecnologia que melhor será aplicada, são algumas delas, capacidade de carga, quantidade de sensores, quantidade de horas ininterruptas para os que possuem bateria, compatibilidade com a planta instalada, qual tipo de tarefa a ser realizada, entre outros.

3.2.1 AGV Guiados Por Trilhos Fixos e Móveis

Veículo automatizado cujo modelo é o mais simples se tratando de tecnologia, pois trafega apenas em linha reta sobre um trilho montado por todo seu percurso. Possui sensores ópticos ou chaves fim de curso para indicação de pararem com segurança, conforme figura 1.

Segundo Leuze eletronic (2018) “o monitoramento é feito sobre a sua velocidade e da sua posição. Com sensor de medição a laser localizado no final do percurso mirando para o veículo, transmitindo em tempo real todas as informações para um sistema.”

Sua utilização é melhor aplicada em plantas simples e que não exijam a necessidade de um AGV guiado automaticamente, sua principal vantagem é o preço para implementação.

Figura 1 – AGV Guiado por Trilhos
Fonte: http://pt.automtransfer.com/ (2020)

3.2.2 AGV Fita Magnética

Este AGV possui um conjunto de sensores na parte inferior do veículo que detectam a presença de uma fita magnética que enviam sinais para um controlador que automaticamente efetua o redirecionamento AGV fazendo com que ele permaneça a todo o momento sobre a trilha. Sua utilização é melhor aplicada em pisos irregulares com muita sujeira e com intenso fluxo, conforme figura 2.

Figura 2 – AGV Guiado por Fita Magnética
Fonte: https://www.turck.us/ (2020)

3.2.3 AGV Fita de Alto Contraste

Este modelo é similar ao veículo guiado por fita magnética com uma diferença quanto a tecnologia usada para guiar esse AGV, o funcionamento básico desse modelo na detecção de uma fita ou marcação de alto contraste, seja ela pintada ou alto relevo. Os sensores ópticos que irão detectar a fita de alto contraste (normalmente mais escura que o piso a ser aplicada) e enviarão ao controlador que efetua automaticamente a redirecionamento fazendo com que ele permaneça sobre a trilha. Sua utilização é melhor aplicada em pisos regulares com pouca sujeira e com pouco fluxo, conforme figura 3.

Figura 3 – AGV Guiado Por Fita de Alto Contraste
Fonte: https://www.sparkag.com.br/ (2020)

3.2.4 AGV Guiado Por Laser

Este modelo é equipado com um sensor óptico que emite um sinal laser que reflete nos receptores instalados ao longo do trajeto. e é lido novamente pelo controlador para orientá-lo. Este AGV emite um sinal que reflete nos receptores que é lido novamente enviando-o para o controlador, a diferença entra emissão e captura do sinal é calculado e enviada ao controlador permitindo saber sua localização e se necessário corrigir sua posição e trajetória. Esta tecnologia é melhor aplicada em plantas onde é inviável ou não é possível a instalação do AGV guiado por fita magnética ou fita de auto contraste, conforme figura 4.

Figura 4 – AGV Guiado a Laser
Fonte: http://pt.automtransfer.com/ (2020)

4 COMPONENTES UTILIZADOS

4.1 Arduíno

A plataforma de desenvolvimento Arduíno foi criada e desenvolvida na Itália em 2005, através de um grupo de 5 pesquisadores com um intuito de viabilizar projetos de aprendizagem em escolas e instituições, pois até então, nem todas as instituições tinham condições de ter equipamentos adequados devido aos altos custos. Diante disso decidiram criar uma placa de baixo custo com as mesmas características de um computador para otimizar o aprendizado de seus alunos.

No intuito de aumentar mais ainda o aprendizado com o Arduíno foram criados módulos de expansão, como sensores de temperatura, umidade, encoders para motores entre outros, que são conectadas as entradas e saídas existentes no Arduíno. Pelo fato do Arduíno original ser *Open Source* (Código Aberto em português), ou seja, se o módulo for desenvolvido por outras pessoas, mas utilizando a mesma linguagem e elaboração, este irá realizar as mesmas ações dos originais.

Essa ferramenta não ficou apenas em instituições, muitos artistas, designers ou qualquer pessoa apaixonada por robótica começaram a comprar para criar projetos particulares dando origem o termo “faça você mesmo”.

4.1.1 Funcionamento

Para o funcionamento do Arduíno é preciso ter no mínimo entrada(s), saída(s) e uma programação carregada em sua memória.

Essas entradas podem ser por meio de sinais de sensores, botoeiras, módulos entre outros, sendo elas: sinais digitais ou analógicos, que são conectadas através de terminais no corpo do Arduíno e essa quantidade de terminais de entrada e saídas dependem da versão do modelo do Arduíno a ser utilizado. As saídas também utilizam sinais digitais ou analógicos capaz de alimentar reles, lâmpadas, e uma infinidade de “*Shiled’s*” cada um com uma função em particular, dependendo do projeto a ser realizado.

Na sua programação o Arduíno compreende as variáveis recebidas pelas entradas e de acordo com essa programação irá acionar as saídas.

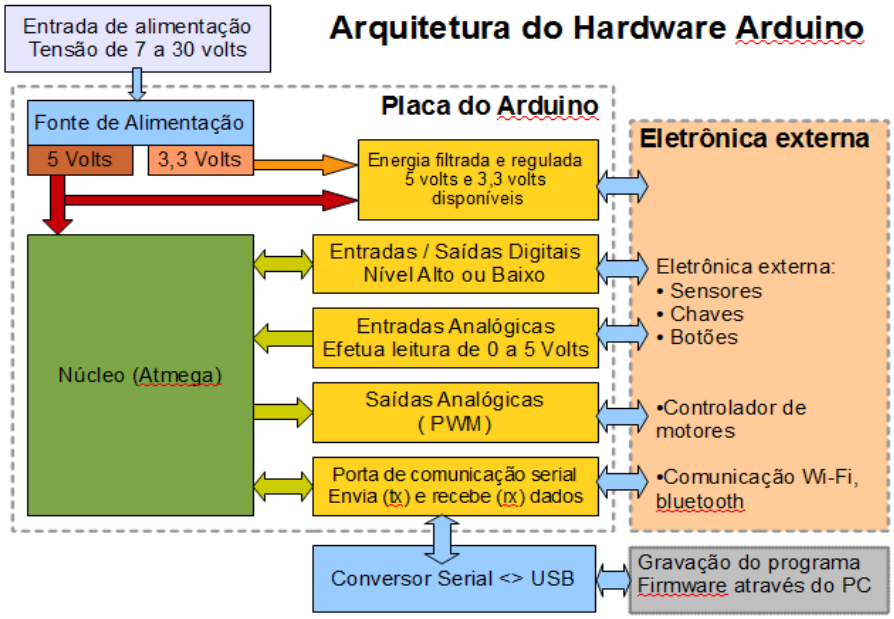

Figura 5 – Arquitetura do *Hardware* do Arduíno MEGA 2560
Fonte: https://athoselectronics.com (2020)

4.1.2 Arduíno Mega

Nesse projeto foi usado o Arduíno Mega 2560 que possui um total 70 portas, sendo elas, 54 entradas e saídas digitais, 15 podem ser usadas como saídas PWM, 16 entradas e saídas analógicas, 4 usadas como comunicação serial. Esse modelo oferece também uma memoria 800% superior ao Arduíno padrão (Uno), seu microcontrolador Atmega2560 proporciona maior quantidade de memória para programação. Os componentes e especificações técnicas, estão ilustradas conforme figura 6.

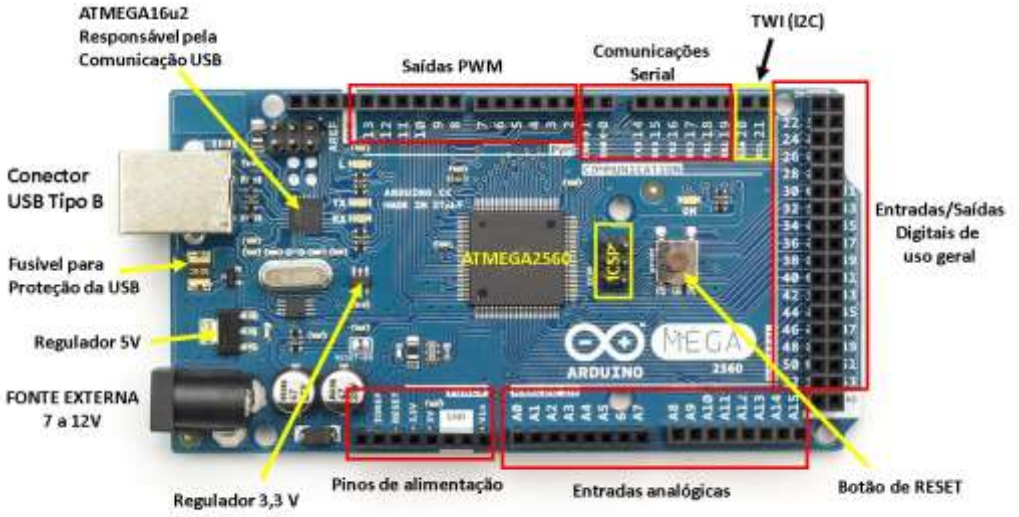

Figura 6 - Resumo de recursos do Arduíno MEGA 2560
Fonte: Embarcados (2020)

- Tensão de operação: 5Vdc
- Voltagem de entrada (Recomendada): 7 a 12Vdc
- Pinos digitais I/O: 54
- Pinos de saída PWM: 14 (Incluso nos 54 digitais)
- Pinos de entrada analógica: 16
- Corrente contínua por pino I/O: 40mA
- Processador: ATMEGA 2560
- Memória flash: 256Kb
- Velocidade de Clock: 16MHz
- EEPROM: 4KB

4.1.3 Software IDE

A IDE (em inglês *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado em português) é um software usado para programação de um projeto usando o Arduino. Esse programa foi desenvolvido para facilitar a utilização por pessoas pouco familiarizadas com a criação de uma programação.

A IDE utiliza a linguagem de programação C e C++. Conta um editor de código com recursos de realce de sintaxe, identificação automática e parênteses

correspondentes, sendo capaz de compilar e carregar a programação para o Arduino. Inicialmente nota-se que existem duas funções obrigatórias para que devem conter em um programa Arduino, setup e loop, conforme figura 7.

- A função *setup*: é executada quando o programa é iniciado e é responsável pelas configurações preliminares do microcontrolador, tais como inicialização da comunicação, declarações das variáveis, definição dos pinos de I/O, entre outros.
- A função *loop*: onde acontece as rotinas repetidas infinita vezes da programação, ou seja, onde as instruções inseridas serão executadas continuamente pelo microcontrolador. (EMBARCADOS 2020).

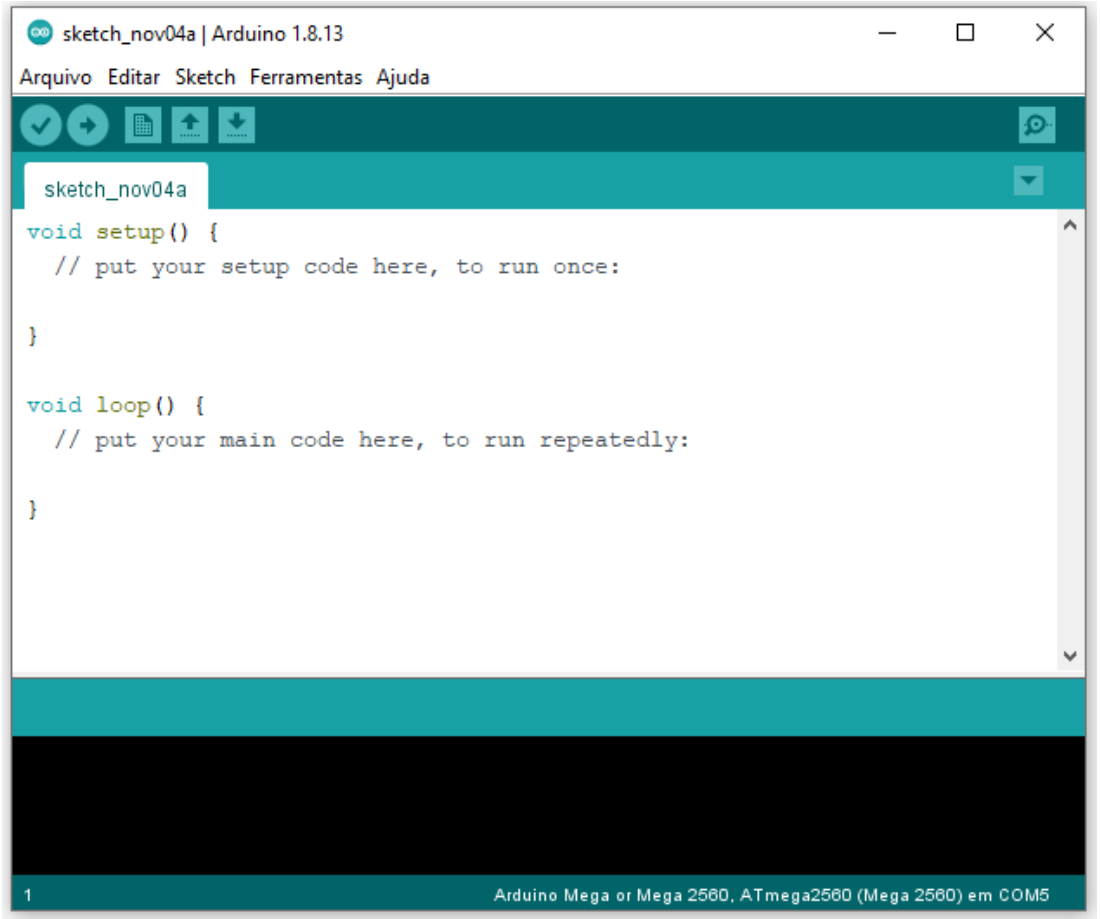

Figura 7 – Tela de programação Arduíno IDE
Fonte: Arduino IDE (2020)

4.2 Controlador de Velocidade para Motores DC

4.2.1 Circuito Integrado de Potência Para Motores DC

Os circuitos integrados de potência são usados para controlar motores de corrente contínua e outras cargas bidirecionais. Estes integrados são capazes de controlar o sentido de rotação de motores a partir de sinais lógicos de microcontroladores e outros circuitos, sendo que, em alguns casos controlam a velocidade ou potência usando sinais PWM.

Estes circuitos integrados de potência são construídos para excitar motores DC, e são amplamente usados em shield's comerciais para Arduino.

Uma das vantagens de se ter um circuito assim está relacionado a possissibilidade de uma maior dissipação de potência em relação ao acionamento de forma direta através de portas de circuitos microprocessados, que não conseguem fornecer uma corrente maior que 40mA em suas saídas, com risco iminente de danificar a placa microprocessada, já o circuito integrado é capaz de fornecer até 2A por canal.

4.2.1.1 Circuito Integrado Ponte H

As pontes H têm esse nome devido seu formato que é desenhado de forma didática e se assemelha a letra H. O circuito possui 4 chaves que são usadas para inverter a polaridade sem a necessidade de simétrica conforme figura 8.

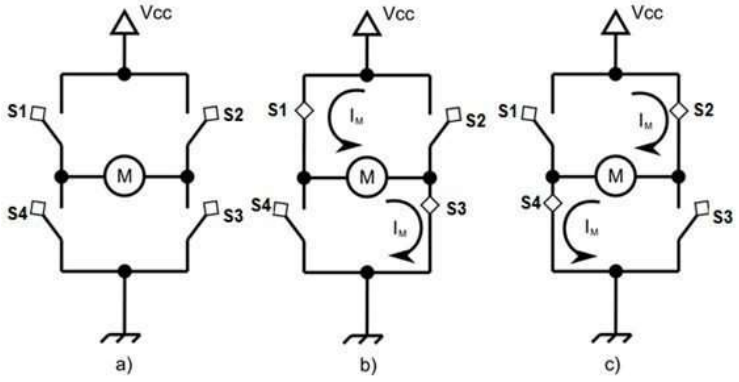

Figura 8 – Funicionamento Ponte H
Fonte: https://portal.vidadesilicio.com.br/ (2020)

O sentido de rotação de um motor de corrente contínua depende do sentido de circulação da corrente através de seus enrolamentos. Isso quer dizer que, invertendo a polaridade de alimentação de um motor, inverte também o seu sentido de rotação.

O funcionamento da ponte H se comporta seguinte maneira: fechando apenas as chaves S1 e S3 a corrente passará em um sentido através dos enrolamentos do motor que irá girar em um sentido, conforme figura 8 (b), fechando apenas as chaves S2 e S4 o sentido do fluxo da corrente que passa através dos enrolamentos do motor é invertido com relação ao anterior, logo o motor gira no sentido oposto, conforme figura 8 (c).

4.2.1.2 Circuito Integrado L298P

O chip L298P é utilizado para controlar sentido da rotação e a velocidade de motores de corrente contínua utilizando a comutação de suas chaves internas que são realizadas eletronicamente através de duas entradas independentes conforme diagrama abaixo.

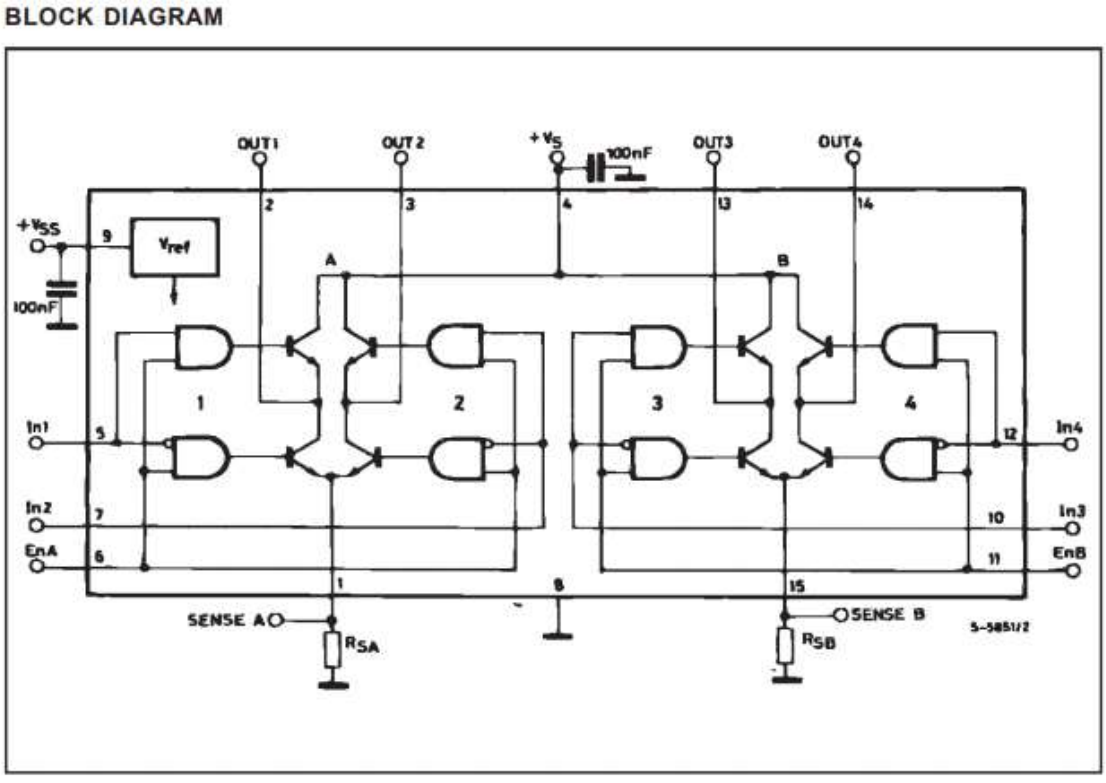

Figura 9 – Diagrama de Bloco L298P
Fonte: https://www.alldatasheet.com/ (2020)

Esse controlador possui dois circuitos H, dessa forma, é capaz de controlar dois motores.

O controle de velocidade do chip L298P é feito utilizando sinais PWM, Pulse Width Modulation (Modulação por Largura de Pulso), basicamente, consiste em aplicar uma onda quadrada de amplitude de tensão e frequência alta no lugar da tensão continua, sendo portanto, possível regular a tensão de saída, e dessa forma regular a velocidade dos motores.

4.2.1.3 Shield Arduino - Motor Drive

O Arduino Shield Motor Drive é uma placa desenvolvida para ser utilizada com o Arduino com o intuito de controlar a velocidade e o sentido de rotação de motores de corrente contínua de maneira independente. Este shield usa o chip L298P para controlar até 2 motores DC de 7,5V a 20V com máximo de 2A de corrente conforme figura abaixo.

Figura 10 – Arduino *Shield Motor Drive*
Fonte: https://www.robocore.com/ (2020)

Especificações:

- Tensão da lógica: 5V (vem do Arduino)
- Tensão que pode ser drenada aos motores: 7,5V a 12V (através do pino VIN) ou 7,5V a 20V (através de fonte externa - selecionável via jumper)
- Controle de velocidade em ambos os sentidos de maneira independente

- Até 2A de corrente para cada motor
- Pinos usados para mover 2 motores: ~5, ~6, 7 e 8 (~ indica PWM)
- Controle de velocidade via PWM
- Bornes destacáveis para facilitar as conexões

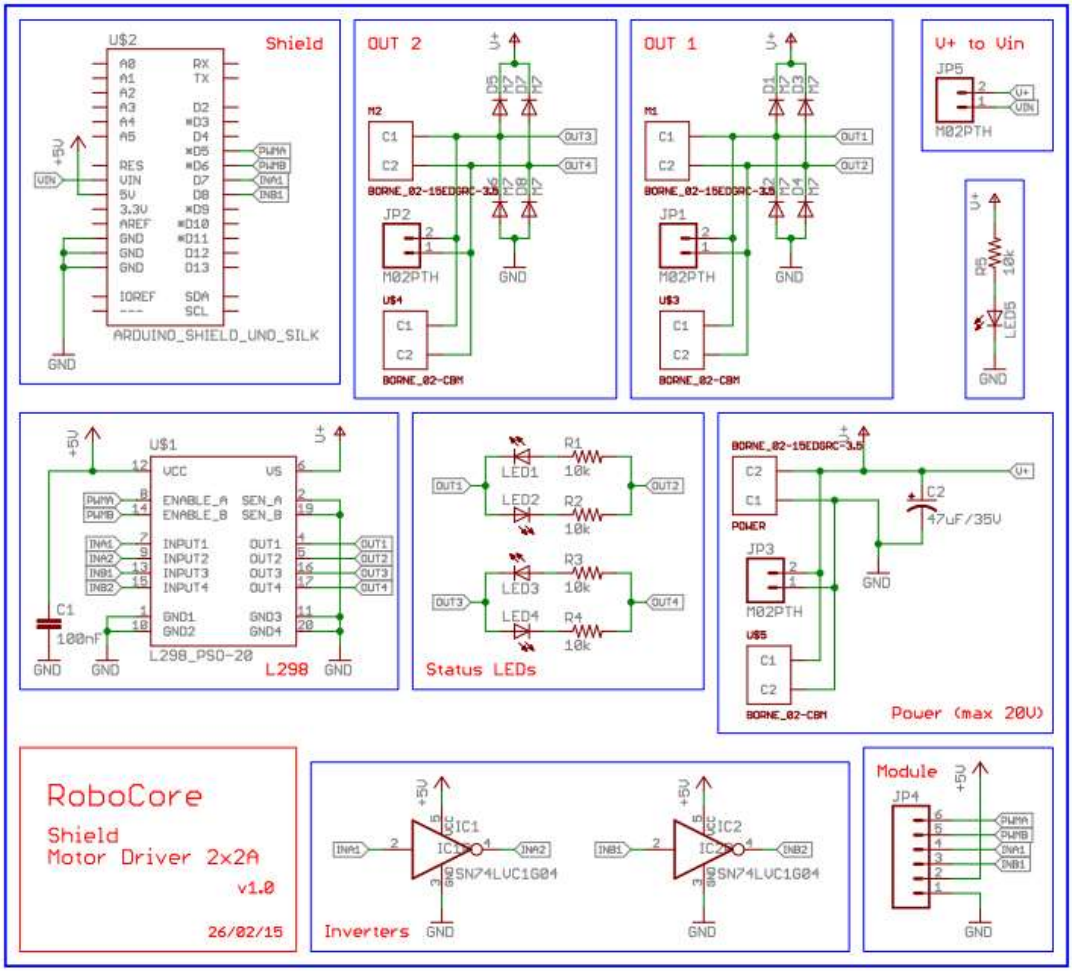

Figura 11 – Diagrama de Blocos *Shield Motor Drive*
Fonte: https://www.robocore.com/ (2020)

4.3 Sensor Ultrassônico

4.3.1 Definição

As ondas ultrassônicas são ondas sonoras com frequências além da capacidade da audição humana, ou seja, ondas acima de 20kHz.

Sensores Ultrassônicos geram ondas ultrassônicas que são transmitidas na forma de um cone conforme figura 12, que no momento que interceptam a superfície

de um corpo é refletida de volta ao sensor transmitindo informações a respeito da velocidade de deslocamento e a distância do sensor. Essas informações são geradas da diferença entre as frequências das ondas emitidas e recebidas, e do intervalo de tempo entre a emissão e o retorno da onda. A propagação e reflexão das ondas podem ser afetadas por algumas variáveis causando distúrbios a detecção do objeto como, mudança de temperatura ou humidade, aspereza da superfície, incluindo o formato, que afeta a quantidade de sinal sonoro refletido por exemplo em objetos arredondados comprometendo o retorno da onda ultrassônica.

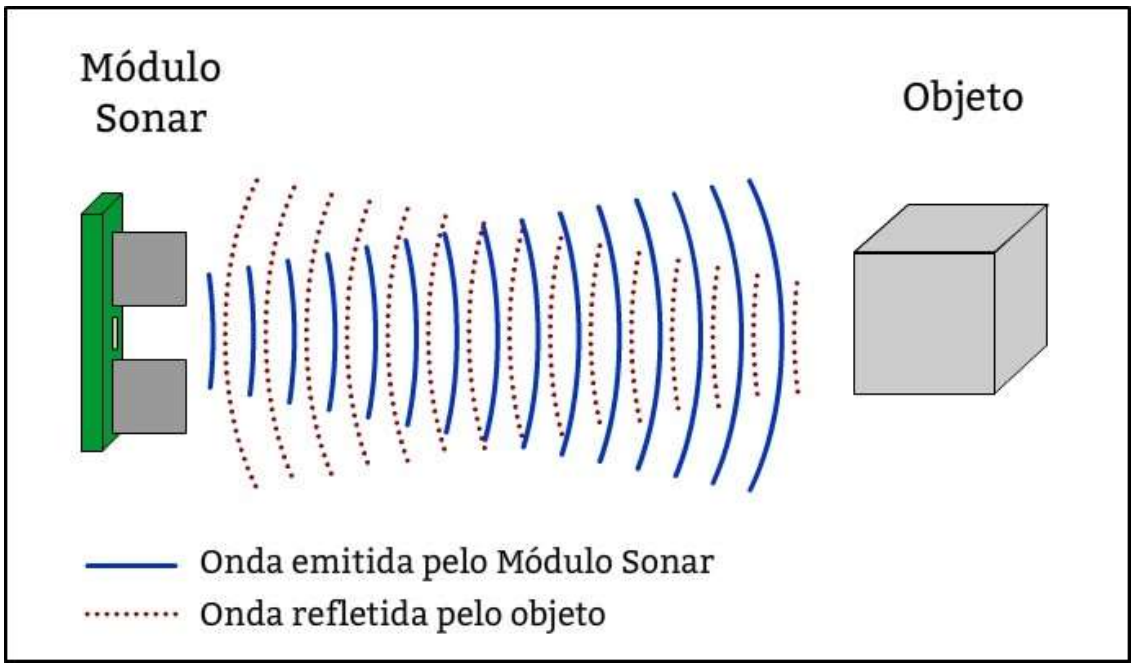

Figura 12 – Transmissão e reflexão da onda ultrassônica
Fonte: https://www.ricardoteix.com/ (2020)

4.3.2 Sensor HC-SR04

O sensor ultrassônico HC-SR04 (Figura 13) utiliza ondas ultrassônicas para determinar a distância de um determinado objeto sendo capaz de detetá-lo sem contato físico, conforme figura 14, com alta precisão e leituras estáveis e dinâmicas. Seu modo de operação não é afetado por material na cor preta ou pela luz solar. Usado para medir distâncias entre o sensor e um objeto, também sendo possível configurá-lo inserindo uma instrução na programação para acionar uma saída ao detectar um objeto a uma distância pré-programada, entre muitas outras aplicações.

Figura 13 – Sensor Ultrassônico HC-SR04
Fonte: https://create.arduino.cc/ (2020)

Características:

- Tensão de Alimentação: 5Vdc
- Corrente de trabalho: 15mA
- Ângulo efetivo: <15º
- Alcance: 2cm - 400cm / 1” - 13 pés
- Erro: 0.3cm
- Ângulo de medição: 30º
- Frequência de operação: 40KHz
- Largura do Pulso de entrada do acionador (Trigger): 10uS
- Dimensão 45mm x 20mm x15mm

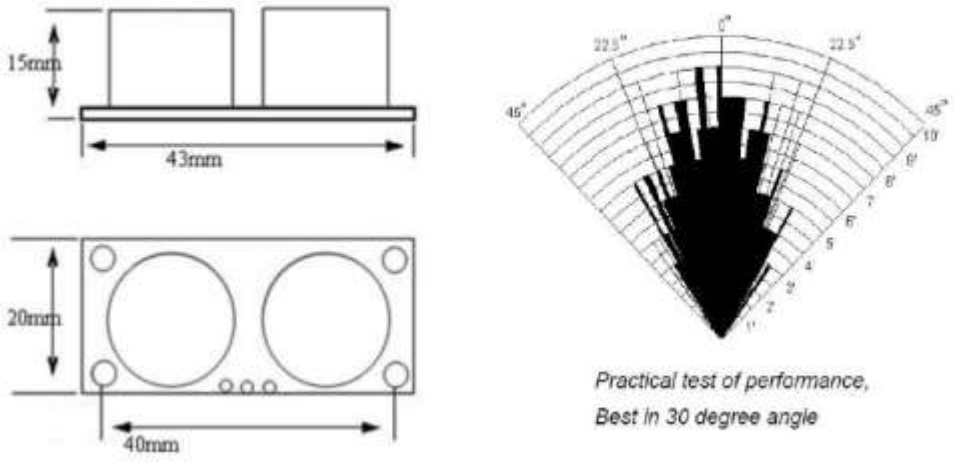

Figura 14 – Amostra de performance e ângulo de operação HC-SR04
Fonte: Cytron Technologies (2020)

4.3.3 Funcionamento

O módulo do sensor possui 4 pinos de conexão, dos quais 2 são para alimentação, um para iniciar a medida (pino "*Trigger*") e outro para o resultado da medida (pino "*Echo*") conforme figura 15. Seu funcionamento é composto por uma lógica, o módulo recebe um sinal de largura de 10 µs ou mais, no pino *Trigger*, emite um sinal sonoro composto de 8 ciclos de 40 kHz no transmissor Tx e aguarda o retorno desse sinal no receptor Rx. O intervalo de tempo entre a emissão do sinal e o seu retorno constitui a largura do sinal gerado no pino Echo, conforme figura 16. Com posse da velocidade de propagação do som, pode-se converter o tempo em distância, esta conversão é realizada dentro do hardware do Arduíno, através do código carregado.

Figura 15 – Pinos de Conexão Sensor Ultrassônico HC-SR04
Fonte: https://create.arduino.cc/ (2020)

Tempo = Largura de pulso do *Echo*, em μs (microssegundos)

- Distância em cm = Tempo / 58
- Distância em polegadas = Tempo / 148
- Ou pode utilizar a velocidade do som, que é 340 m/s

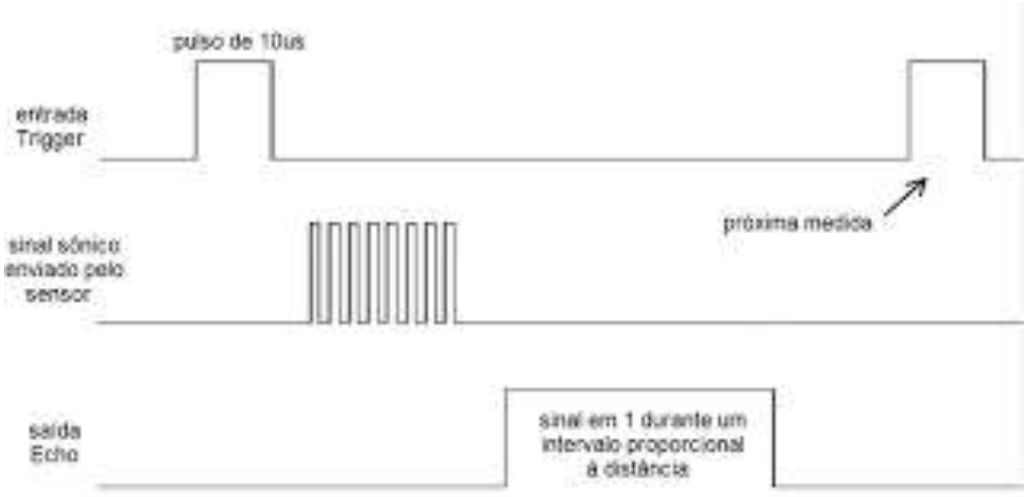

Figura 16 – Pulso e tempo de resposta do HC-SR04
Fonte: Cytron Technologies (2020)

4.4 Sistema GPS

4.4.1 História

Sistema de Posicionamento Global ou Global Positioning System em inglês, mais conhecido pela sigla GPS, é um sistema de navegação eletrônico civil e militar que começou a dar os primeiros passos na década de 60 com o lançamento do primeiro satélite artificial da história, o Sputnik I pela União Soviética.

A partir de inúmeras pesquisas, ficou constatado que os cientistas russos recebiam sinais enviados por este satélite via rádio, este sinal quando recebido em terra emitiria um “beep” que poderia ser ouvido em todo e qualquer rádio em todo globo terrestre que poderiam ser associados com o sistema de coordenadas geográficas para fins de localização.

Mesmo os soviéticos tendo iniciado os estudos sobre a utilização de satélites com finalidade de localização, foram os Estados Unidos que colocaram de fato em prática este sistema, o intuito dos americanos era desenvolver o projeto e se destacar diante da União Soviética, tendo em vista que este era o período em que os dois países travavam uma disputa estratégica.

Baseado nisso, um sistema reverso foi desenvolvido com o intuito de determinar a localização exata de um ponto em toda superfície terrestre sendo criado o sistema *Transit* também em 1967 que futuramente veio a apresentar alguns tipos de limitações quanto a sua precisão, tendo que passar por diversos avanços até alcançarmos o sistema *Navstar* que atualmente vem sendo utilizado em todo mundo.

O projeto criado pelos estados Unidos utilizando um sistema reverso foi intitulado de Transit, com a finalidade de determinar uma localização exata em qualquer ponto da superfície terrestre. Com o surgimento de algumas limitações relacionadas a precisão da localização, esse sistema então passou por inúmeros avanços, até chegarmos no sistema NAVSTAR desenvolvido pelo Departamento de Defesa do país e que é usando atualmente em todo mundo.

4.4.2 Coordenadas Geográficas

Para que o sistema GPS funcione corretamente, é indispensável estabelecer alguns conceitos geográficos para padronizar o processo de determinação de

localização de pontos na superfície da terra. A Terra pode ser simplificada como uma esfera com um ponto localizado no seu centro e raio médio do círculo equatorial aproximado de 6370 km.

A Terra gira em torno de seu eixo, que corresponde à reta que hipoteticamente atravessa a superfície terrestre no pólo sul e pólo norte. Cada plano contendo seu eixo é chamado de plano meridiano e sua intersecção com a superfície esférica aproximada da Terra é denominado de círculo meridiano (ou, apenas, meridiano). O meridiano principal, usado como referência para os outros, é o meridiano de Greenwich, que passa pela cidade de Londres, Inglaterra. A partir deste, define-se como longitude o ângulo λ entre um meridiano qualquer com o meridiano de Greenwich, alternando de 0 h a 12 h (ou 0º a 180º) para leste ou oeste, positivo ou negativo, respectivamente.

Além disso, existem alguns círculos paralelos, que são produzidos pela intersecção de um plano perpendicular ao eixo da terra com a superfície esférica terrestre. O plano que gera o maior paralelo possível é denominado de plano equatorial e este paralelo denominado Equador, o qual divide a superfície terrestre em dois hemisférios: boreal (ou norte) e austral (ou sul). Com isso, é definida a latitude como o ângulo ϕ entre o vetor normal a um ponto terrestre com o plano equatorial. Apresenta valores positivos ou negativos para o norte ou sul de 0º a 90º, respectivamente, a partir do linha do Equador, conforme Figura 17, ilustra a Terra com seus principais paralelos e meridianos, além de uma posição P dada em latitude e longitude.

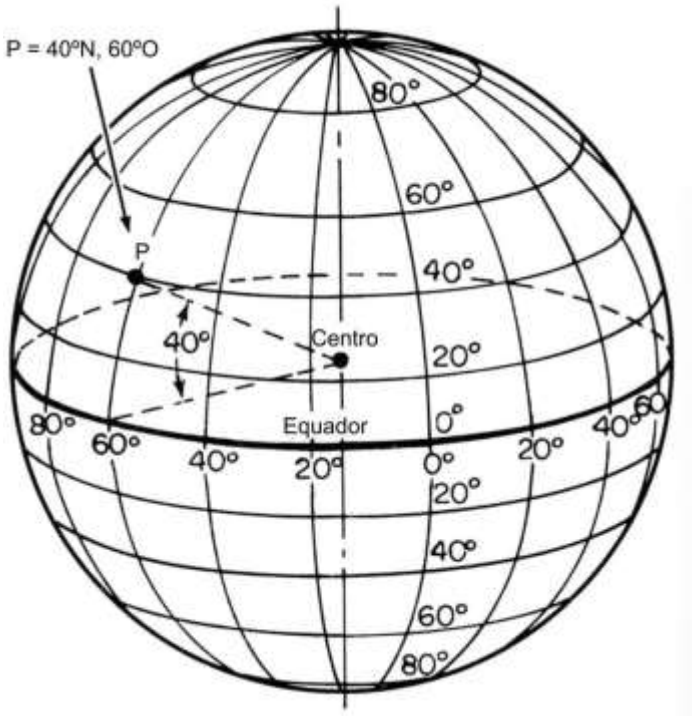

Figura 17 – Globo Terrestre com alguns paralelos e Meridianos
Fonte: Cytron Technologies (2020)

A longitude λ e latitude ϕ medidas em graus, junto com a altitude h medida em metros (a partir do nível do mar), correspondem às grandezas de maior utilização na Geografia. Entretanto, o sistema de coordenadas cartesiano mapeado pelos eixos x, y e z são usados mais matematicamente. Portanto, a maioria dos modelos e ferramentas usados nas diversas áreas da engenharia utiliza este sistema de coordenadas como padrão. Desta forma, o sistema de coordenadas ECEF (“earth-centered earth-fixed coordinate system”) é utilizado para representar pontos referentes à Terra por dos eixos do sistema cartesiano. O sistema ECEF contém sua origem no centro da esfera referente à Terra, com o plano xy posicionado sobre o plano equatorial (longitude de 0º e +90º nos eixos x e y, respectivamente) e eixo z acima do eixo de rotação terrestre direcionados para o norte com valores positivos, sendo, portanto, o sistema ECEF fixo em relação à rotação terrestre. As coordenadas geográficas são indicada por dois ângulos: o longitudinal e o latitudinal. O ângulo longitudinal é em relação à linha imaginária de Greenwich e o ângulo latitudinal é em relação à linha imaginária do equador.

4.4.3 Sistema Navstar GPS

O Sistema utilizado atualmente é proveniente da fusão dos sistemas (Time Navegation) vindo da marinha norte americana e do sistema System 621- b, este vindo das forças aéreas. Após esta fusão, o sistema de navegação passou a ser chamado de Navstar-GPS (Navegation System with timing e raning).

Como dito anteriormente o Navstar teve origem no departamento de defesa norte americano (USDod), visto que as forças armadas necessitavam de maior precisão de posicionamento, algo que os meios desenvolvidos anteriormente não alcançavam êxito. Porem, visto a necessidade, este sistema que inicialmente era utilizado apenas pelas forças armadas passou a ser utilizado em demais ramos da sociedade em geral. Basicamente os três segmentos são distintos e operam em constante interação tendo cada um suas funções específicas, processando as informações e proporcionando simultâneo e continuamente, dados de posicionamento tridimensional determinando uma localização com alta precisão.

Figura 18 - Distribuição de satélites em torno da terra
Fonte: Dreamstime (2020)

4.4.3.1 O segmento espacial (satélites)

A parte espacial de todo esse processamento é constituídas por 27 satélites sendo que 24 estão diretamente em uso enquanto os demais ficam de reserva, todos estes 27 ficam a aproximadamente 20200 Km da superfície terrestre, com um período de 12 horas siderais sendo totalmente distribuídos em 6 orbitas separados entre si em 60° com uma angulação em relação ao plano equatorial de 55°. Todo este sistema de posicionamento destes satélites foi rigorosamente estudado a fim de obter uma maior precisão neste posicionamento, sendo que nesta distribuição em todo e qualquer ponto da superfície este sistema garantirá em uma probabilidade de 95% que 4 destes satélites estejam visíveis acima do horizonte.

Figura 19 – Satélites em torno do receptor
Fonte: http://vaztolentino.com/ (2020)

O sistema de referência associado ao GPS, quando se utilizam efemérides transmitidas, é o WGS 84. Desta forma, quando um levantamento é efetuado usando o GPS na sua forma convencional, as coordenadas dos pontos envolvidos serão obtidas nesse sistema de referência. Sua origem é o centro de massa da Terra, com os eixos cartesianos X, Y e Z

4.4.3.2 O segmento terrestre (monitoramento e controle)

Sua principal função é manter o processamento de informações transmitidas pelos satélites em tempo real. São 5 estações espalhadas pelo mundo, localizadas em Kwajalein, Ascencion, Colorado Springs, Diego Garcia, Hawaii, que continuamente rastream todos os satélites visíveis as antenas das estações pelas Monitoring Stations (MS) e são transmitidos para a Master Control Station (MCS) em Colorado Springs, nos estados Unidos.

Figura 20 – As 5 centrais de controle distribuídas no globo
Fonte: http://vaztolentino.com/ (2020)

4.4.3.3 O segmento do usuário (receptores GPS e equipamentos associados)

O segmento de usuário são os equipamentos receptores transmitidos de forma continua pela constelação de satélites GPS. O segmento é composto por equipamentos espalhados na superfície do globo e também estão presentes em automóveis, aviões e embarcações marítimas, além de aparelhos móveis projetados para diversas finalidades distintas, como celulares, computadores, relógios, entre outros.

São todos os equipamentos relacionados a comunidade de usuários que utilizam para determinar a posição, tempo ou até mesmo a velocidade. Os recepctores GPS são construídos basicamente de um pré amplificador, uma antena e uma unidade onde são integrados os componentes eletrônicos usados no controle, registro e

visualização dos dados.

Cada receptor pode ser projetado para uma finalidade de aplicação que pode ter acesso a serviço disponível aos civis pelo Código C/A ou até um equipamento exclusivo para serviço militar através do Código P com maior precisão.

Existe uma grande variedade nos tipos e modelos de receptores GPS existentes nos tempos de hoje, tanto no âmbito das antenas usadas para a recepção das ondas eletromagnéticas emitidas pelas antenas dos satélites quanto do restante do *hardware* responsável por amplificar esses sinais, decodificando informações das efemérides e executando processamento das mesmas.

Figura 21 – Visão Geral de todos os segmentos que constituem o GPS
Fonte: http://ead.senar.org.br/ (2020)

4.4.4 Posicionamento

O posicionamento do GPS é determinado pela distância entre o ponto receptor e o ponto usado como referência, que no caso, são os satélites que possuem sua localização conhecida.

Um sinal é emitido pelo satélite e recebido pelo receptor, a medição é feita nesse intervalo de tempo, como já dito, cada receptor está rodeado de pelo menos 3 ou 4 satélites posicionados em sua órbita, o receptor receberá continuamente sinais dos satélites de pontos diferentes, com variação no intervalo de tempo de acordo com cada posição dos satélites.

Com essa medição relativa do tempo são definidas as distâncias automaticamente, determinando assim, a velocidade exata que esses satélites transmitem esse sinal, com isso obtêm-se a relação que:

$$v_m = \frac{\Delta s}{\Delta_v} \quad (1)$$

Onde:

Vm = Velocidade Média em que os satélites transferem o sinal via rádio.
Δs = Variação da distância entre ponto receptor e o ponto de referência (Satélite)
Δv = Variação de tempo entre o envio e recebimento do sinal.

4.5 Motores

Na atualidade é possível encontrar Motores elétricos em diversos lugares, pois sua combinação de vantagens como construção simples, simplicidade de comando, fácil limpeza e por utilizar energia elétrica que tem um custo baixo, os motores têm uma versatilidade em diversas aplicações e tipos de carga.

A principal função de um motor elétrico é transformar energia elétrica em energia mecânica, essa energia elétrica pode ser dividida em alternada e contínua.

4.5.1 Motores CC

Nosso projeto utiliza um motor de corrente contínua, onde sua energia elétrica é fornecida por uma bateria de 9V e seu comando vem das saídas do Arduino.

O funcionamento desse motor é simples, onde no seu estator (parte fixa do motor) é fixado pares de imãs que aponta seu lado norte e o outro sul para o eixo do motor, nesse eixo é localizado as bobinas que sempre terão um número par e sua alimentação é conduzida por escovas como mostra a figura 22. Ao fornecer corrente elétrica através das escovas para as bobinas, é gerada uma força magnética que

produz torque que faz o motor girar. No motor de corrente contínua é necessário apenas inverter os fios positivo e negativo para mudar o sentido que o motor gira e para controlar sua velocidade necessita apenas regular sua tensão.

Figura 22 – Componentes de um motor
Fonte: Citisystems (2020)

5 ESTUDO DO CASO

5.1 Desenvolvimento

Com posse de todos os componentes do protótipo, foi dado início a etapa de montagem, comparações e testes.

O protótipo foi fragmentado em pequenas partes, dividindo-o por função, ou seja, de forma individual. Iniciando pelos testes com o sistema de sensor de obstáculos, posteriormente com o sistema de movimentação e por último o sistema de coordenadas usando módulo GPS. Com todos os sistemas preparados e devidamente testados foi iniciada a integração de hardware e software.

Ao longo desse capítulo serão descritas as dificuldades encontradas durante a realização deste trabalho, soluções tomadas e melhorias futuras.

A programação do projeto feita no arduino usando a plataforma IDE que se encontra no anexo 1 desta monografia.

5.1.1 Protótipo

O desenvolvimento do protótipo foi realizado basicamente com o estudo do funcionamento de cada dispositivo de forma individual, incluindo os testes de compatibilidade entre eles, conforme figura 23.

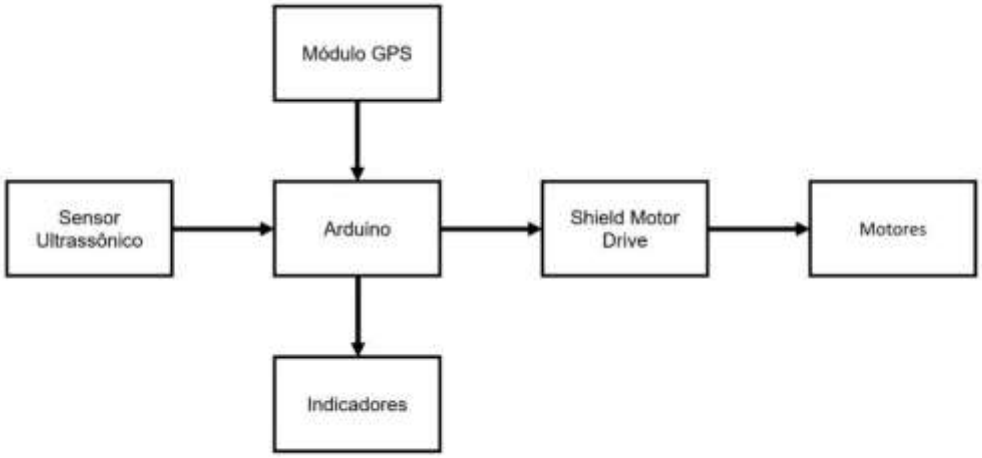

Figura 23 – Diagrama de Blocos do Protótipo
Fonte: Autor (2020)

5.1.1.1 Desenvolvimento do Sistema GPS

Esse teste foi realizado para demonstrar o funcionamento do sistema de GPS. Iniciando pelo teste *"stand-alone"* para verificar o funcionamento do módulo GPS, a fim de entender como são fornecidas as coordenadas e prever qual melhor meio de utilizá-las no protótipo; qual seria o tratamento das variáveis fornecidas na leitura NMEA (National Marine Electronics Association). Após análise, foi verificado que a leitura NMEA era de grande complexidade (informações navais, por exemplo, velocidade em nós) porém, também nos fornecia as coordenadas da forma como se desejava, sendo assim, foi extraído apenas as informações de longitude e latitude.

Em resumo, o módulo GPS recebe os dados de cada satélite, essas informações são extraídas através da biblioteca "*TinyGPS.h*" que faz a conversão dos dados lidos de forma simples, dentre elas, a latitude e longitude. Ele recebe os dados e organiza tudo em modo texto, separando cada dado obtido por ponto e vírgula, desta forma foi possível usar cada uma das coordenadas geradas pelo módulo, conforme tabela1.

Para a aquisição da latitude e longitude foi utilizada a instrução "*gps1.get_position(&latitude, &longitude, &idadeInfo)*" essa instrução armazena nas variáveis "*lat* e *lon"* os respectivos valores e exibe-os no monitor serial com a instrução *"Serial.println(float(latitude) / 100000, 6))"*, o valor 6 indica a quantidade de casas decimais, conforme figura 24.

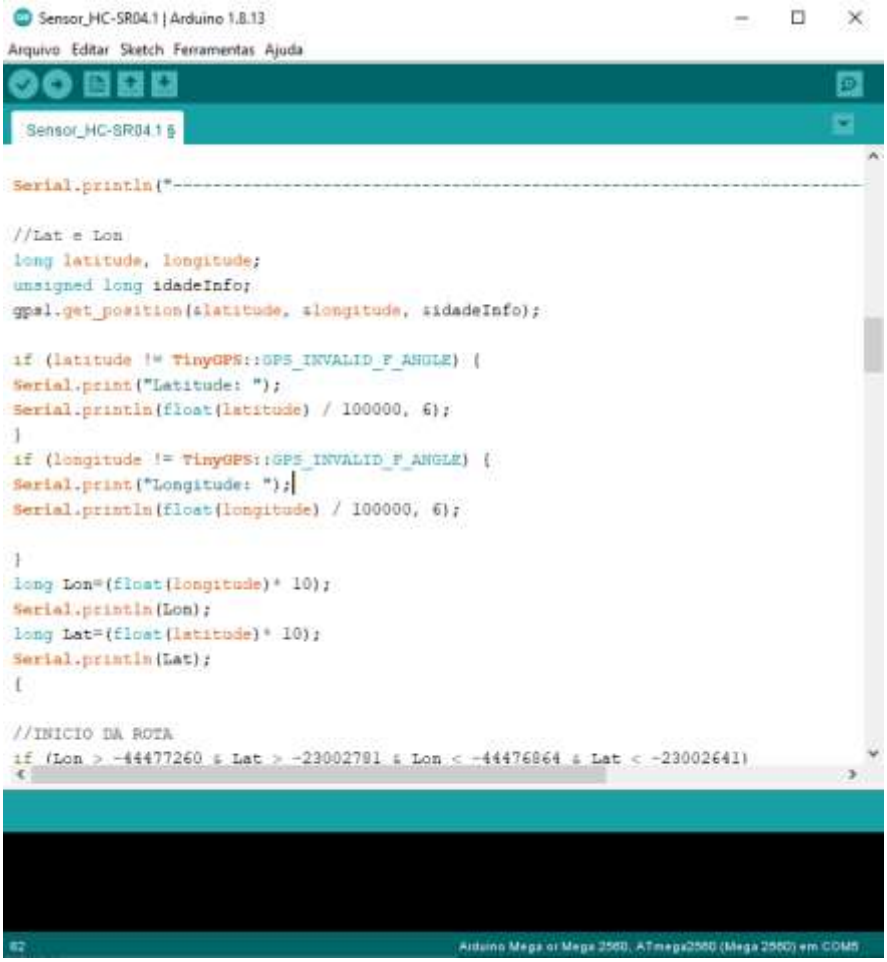

Figura 24 – Trecho da Programação do Módulo GPS
Fonte: Autor (2020)

O sistema GPS foi testado isoladamente, entretanto, era necessário saber quais eram as coordenadas que seriam usadas para realizar a rota do AGV. Ao usar o link do serviço de localização do Google, conhecido como Google Maps, foi possível conseguir as coordenadas, sendo que, a busca da localização desse serviço pode ser realizada por endereços ou por coordenadas, sendo assim foi possível comparar essas coordenadas com as geradas anteriormente pelo módulo, garantindo seu funcionamento.

Com posse de todas as coordenadas da rota desejada foi iniciada a montagem da programação, inserindo os valores correspondentes a cada ponto da rota conforme figura 25. Com isso foi possível determinar qual a rota a ser seguida pelo AGV. Cada ponto mapeado foi determinada uma ação a ser tomada, conforme a tabela 1.

ETAPAS	INTERVALOS DE COORDENADAS		AÇÃO
	Coordenadas Iniciais	Coordenadas Finais	
1	Lon= -44,477260	Lon= -44,476864	Aciona motor I para percorrer a primeira reta do trajeto.
	Lat= -23,002781	Lat= -23,002641	
2	Lon= -44,476864	Lon= -44,476790	Mantém motor I Ligado e aciona motor II até completar a primeira curva do trajeto.
	Lat= -23,002641	Lat= -23,002674	
3	Lon= -44,476790	Lon= -44,476637	Mantém motor I ligado até o final da segunda reta do trajeto.
	Lat= -23,002674	Lat= -23,003013	
4	Lon= -44,476637	Lon= -44,476654	Mantém motor I Ligado e aciona motor II até completar a segunda curva do trajeto.
	Lat= -23,003013	Lat= -23,003068	
5	Lon= -44,476654	Lon= -44,477842	Mantém motor I ligado até o final da terceira reta do trajeto.
	Lat= -23,003068	Lat= -23,003516	
6	Lon= -44,477842	Lon= -44,477900	Mantém motor I Ligado e aciona motor II até completar a terceira curva do trajeto.
	Lat= -23,003516	Lat= -23,003494	
7	Lon= -44,477900	Lon= -44,478060	Mantém motor I ligado até o final da quarta reta do trajeto.
	Lat= -23,003474	Lat= -23,003126	
8	Lon= -44,478060	Lon= -44,478033	Mantém motor I Ligado e aciona motor II até completar a quarta curva do trajeto.
	Lat= -23,003126	Lat= -23,003073	
9	Lon= -44,478033	Lon= -44,477270	Mantém motor I ligado até o final da quinta reta do trajeto.
	Lat= -23,003073	Lat= -23,002791	

Tabela 1 – Mapeamento das Coordenadas em Etapas
Fonte: Autor (2020)

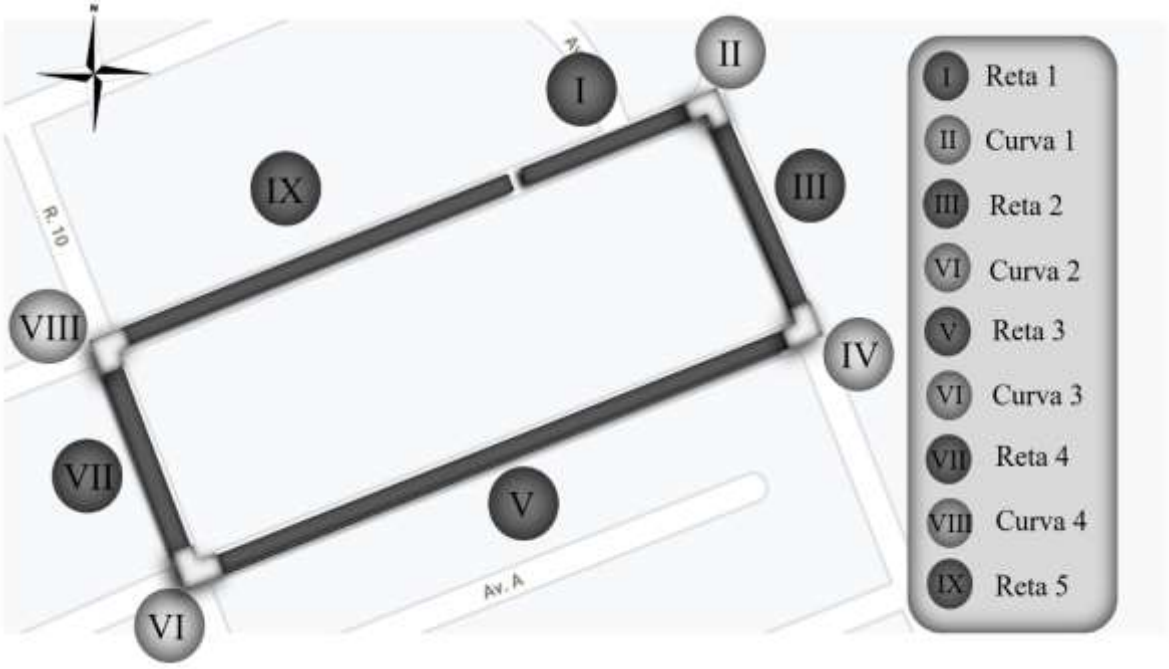

Figura 25 – Mapeamento das Coordenadas em Etapas
Fonte: Autor (2020)

Esse sistema funciona seguindo as etapas conforme a figura 26, o módulo gera a coordenadas e compara as instruções com os valores configurados em cada etapa. Com o reconhecimento das coordenadas da etapa 1, o veículo acelera em direção a etapa 2, e no momento que alcançar as coordenadas configuradas nessa etapa, ele inicia uma curva a direita até alcançar o inicio da etapa 3 ou reta 2, deste modo a programação continua comparando as coordenadas seguindo sua trajetória até o fim do percurso.

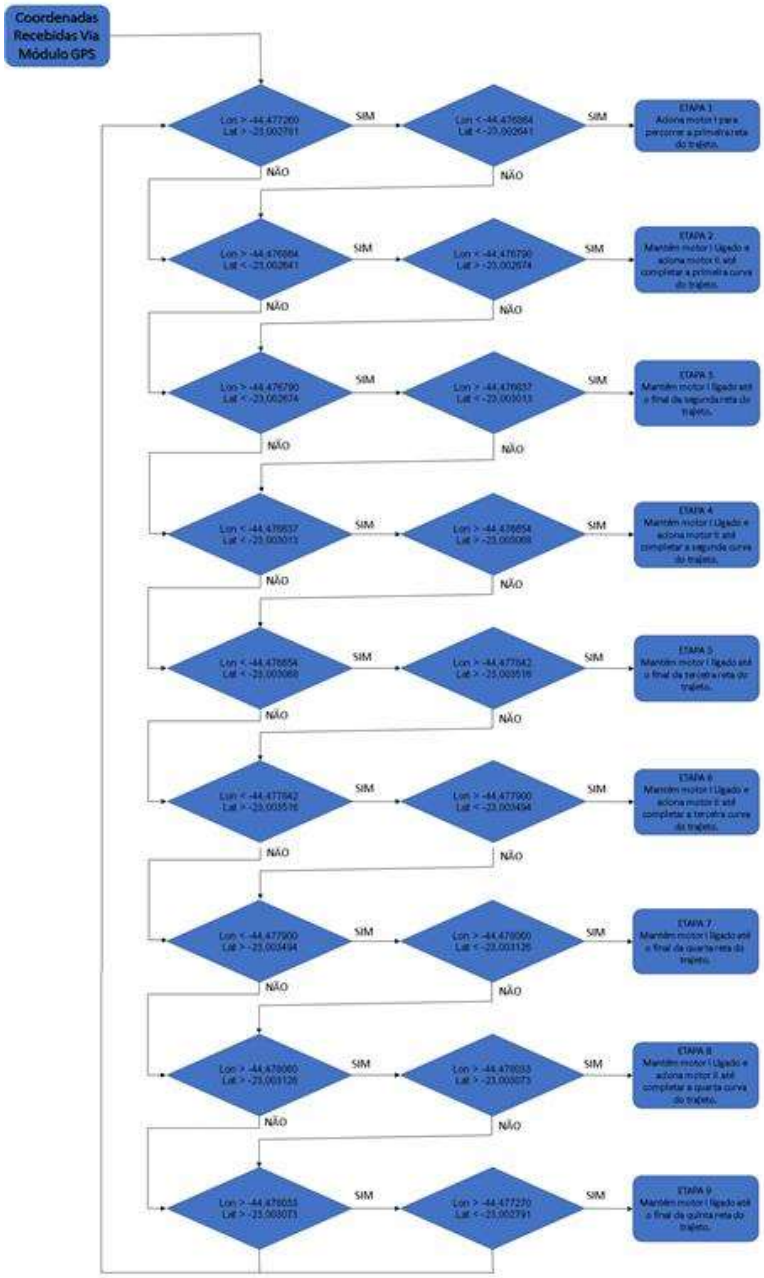

Figura 26 – Fluxograma das Etapas com Coordenadas
Fonte: Autor (2020)

5.1.1.2 Desenvolvimento do Sistema de Detecção de Obstáculo

Esse teste foi realizado para demonstrar o funcionamento do sistema de detecção de obstáculos com diferentes níveis de distâncias usando led's para indicar o momento exato em que eles são detectados pelo sensor.

O sensor dispara um sinal sonoro (ultrassônico) através do pino *Trigger* enviado pela variável inteira *PinTrigger*, esse sinal ultrassônico chega até o objeto e retorna para o sensor em um intervalo de tempo para a pino *Echo* que alimenta a variável *PinEcho* na programação, então, esse intervalo tempo de propagação do sinal é convertido em distância pelo sensor, dessa forma, a programação do sensor foi desenvolvida para acionar um sinal com acionamento de um led na cor azul quando a distância for inferior a 30 centímetros, quando a distâcia for inferior a 20 centímetros e superior a 10 centímetros dispara um segundo led na cor amarela indicando que o obstáculo continua se aproximando e quando a distância for inferior a 10 é desativado a aceleração do motor e acionado um alarme sonoro informando que o obstáculo a frente o impede de continuar sua rota.

No intuito de minimizar erros e/ou paradas desnecessárias, foi inserido três níveis de operação de detecção do sensor, sendo cada um deles representado por led's. A programação foi desenvolvida visando a segurança, portanto, os níveis possuem funções distintas:

- O primeiro nível de detecção é acionado com distância entre o sensor e o obstáculo for inferior a 30 centímetros ativando uma luz de indicação na cor azul, informando que possui um obstáculo a frente, conforme figura 27.

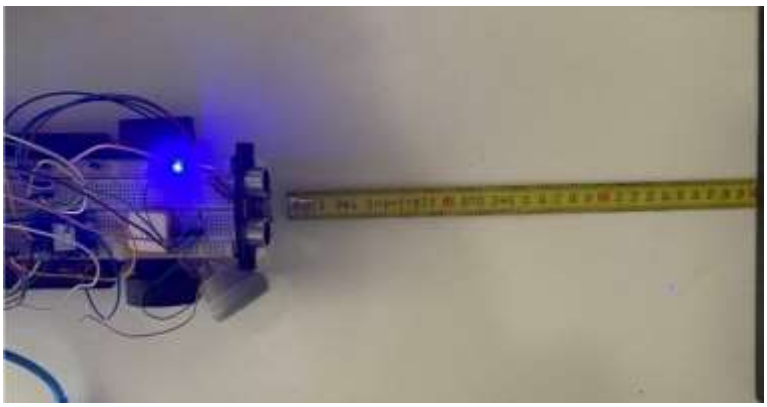

Figura 27 – Teste de Aproximação de Obstáculos 30 centímetros
Fonte: Autor (2020)

- O segundo nível de detecção é acionado quando a distância entre o sensor e o obstáculo for inferior a 20 centímetros ativando uma luz de indicação amarela alertando que o obstáculo continua se aproximando, conforme figura 28.

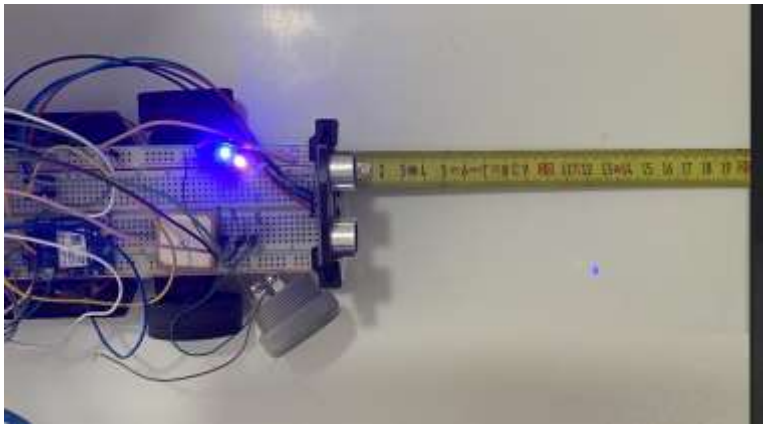

Figura 28 – Teste de Aproximação de Obstáculos 20 centímetros
Fonte: Autor (2020)

- O terceiro nível de detecção é acionado com a distância entre o sensor e o obstáculo for inferior 10 centímetros ativando uma luz de indicação na cor vermelha e acionando alarme de obstáculo a frente desativando a alimentação do motor, conforme figura 29.

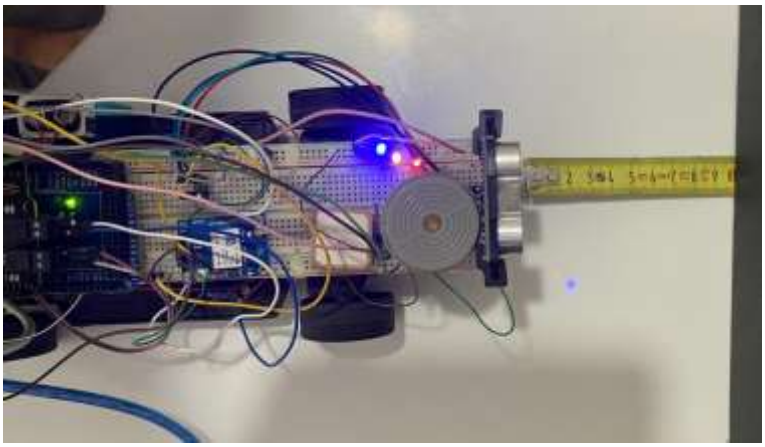

Figura 29 – Teste de Aproximação de Obstáculos 10 centímetros
Fonte: Autor (2020)

Sendo assim, foi observado que, com os níveis de detecção reduziria erros ao cruzar com o obstáculo a frente, minimizando paradas desnecessárias e prejuízos em uma linha de produção. A programação foi desenvolvida conforme figura 30 e o funciona seguindo as etapas conforme figura 31.

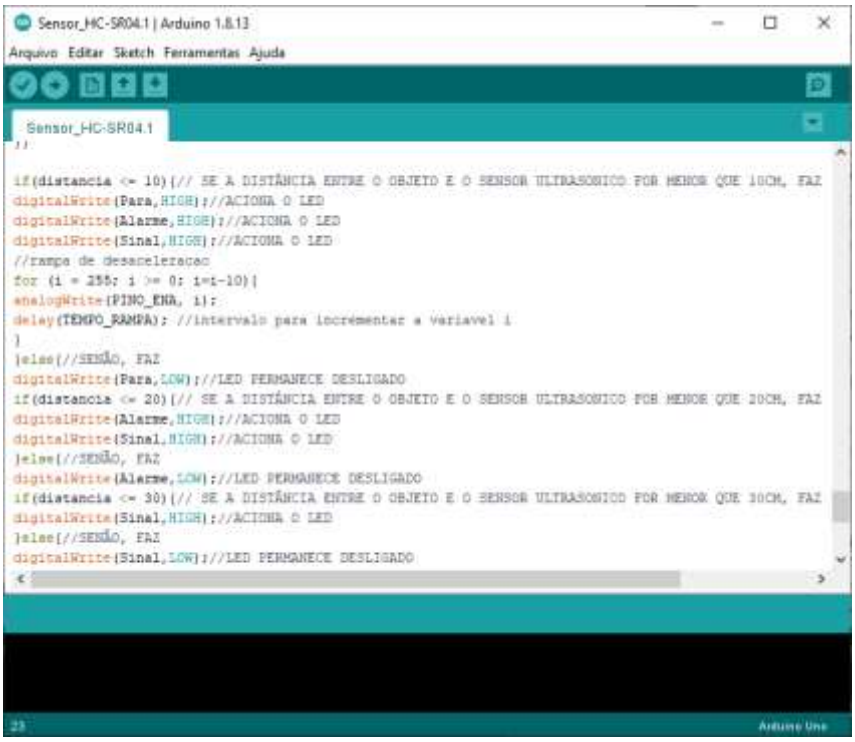

Figura 30 – Trecho da Programação do Sensor Ultrassônico
Fonte: Autor (2020)

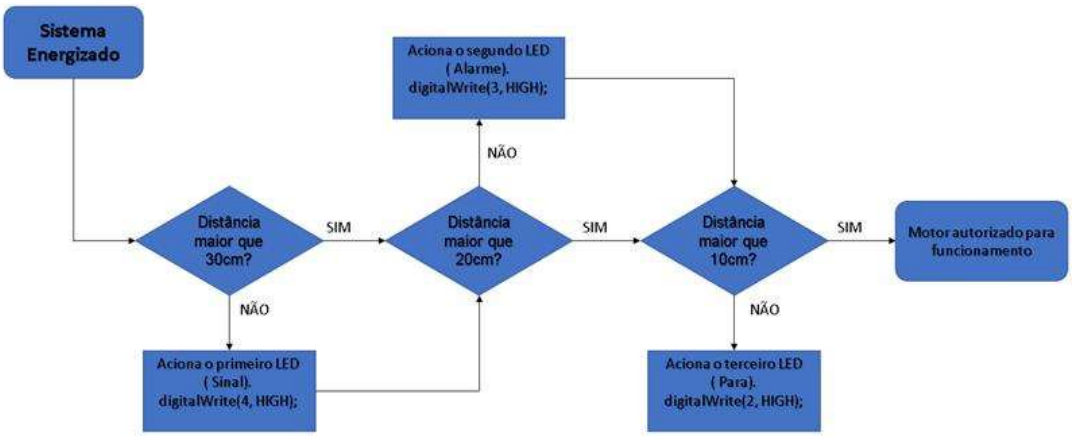

Figura 31 – Fluxograma funcionamento do Sensor Ultrassônico
Fonte: Autor (2020)

5.1.1.3 Desenvolvimento do Sistema Motor

O sistema de movimentação do AGV foi desenvolvido utilizando um chassi de carro de controle remoto em miniatura. Foi utilizado um chassi com as rodas e com o mecanismo de direção para efetuar as curvas pré-programadas, conforme figura 32.

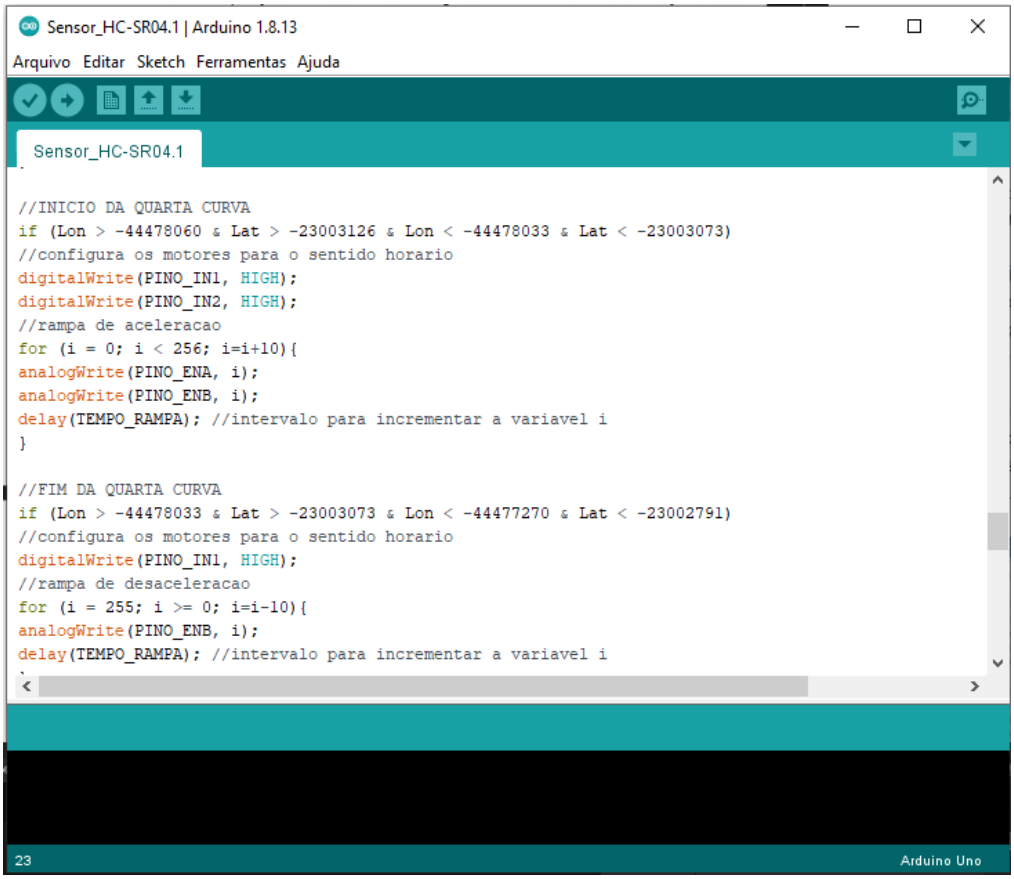

Figura 32 – Trecho da Programação *Shield Motor Drive*
Fonte: Autor (2020)

Os dois motores foram suficientes para a necessidade do projeto, sendo um motor no eixo traseiro para movimentação e um motor no eixo dianteiro para direciona-lo de acordo com a rota, entretanto não se pode conectar os motores diretamente nas saídas do Arduíno, pois pode danifica-las.

Sendo assim, foi necessário a utilização da placa shield motor drive para controlar a velocidade e o sentido de rotação dos dois motores de forma independente, conforme figura 33.

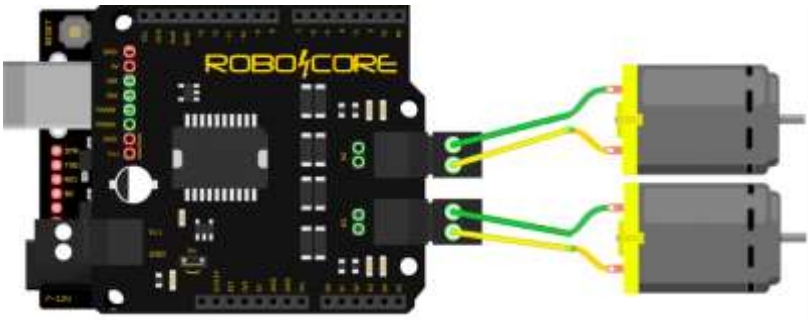

Figura 33 – Ligação do *Shield Motor Drive*
Fonte: Robocore (2020)

5.1.2 Testes de Software do Protótipo

Os testes de software em geral foram realizados de forma incisiva e persistente, tendo que ser revisada inúmeras vezes, nas quais, foram analisados determinados recursos.

A partir desse momento, foi realizado os testes de cada um dos sistemas de formas individual, e foi possível ver o quão complexo seria essa união, o que demandou dias de estudo e várias versões de cada uma das programações.

Um recurso muito utilizado durante o desenvolvimento do protótipo foi o “Serial Monitor” do Arduino. A partir dele, foi possível verificar o que estava acontecendo em tempo real através de mensagens de texto, para que fosse possível acompanhar o processo de funcionamento e ter uma referência das variáveis do nosso protótipo, conforme figura 34.

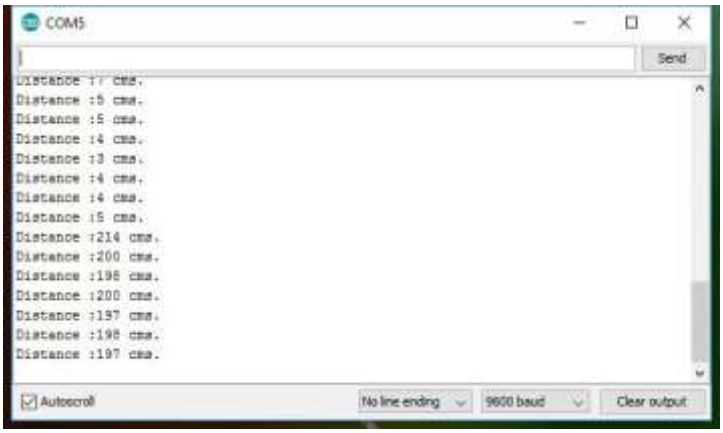

Figura 34 – Serial Monitor Medindo Distancia em Centímetros
Fonte: Autor (2020)

Os testes de detecção de objetos foi o mais simples, pois a programação visava configurar um sensor para reconhecer qualquer obstáculo a frente e tomar a ação de parar o veículo caso se aproximasse a uma distância pré determinada com segurança. Como dito anteriormente, foi inserido níveis de distâncias a fim de minimizar o atraso de resposta do sensor juntamente com indicação por led's. No primeiro nível a programação acende um led na cor azul indicando que existe um obstáculo a frente, o segundo led na cor amarela é acionado indicando que o objeto continua se aproximando, o terceiro led é acionado quando o obstáculo está a aproximadamente 10 centímetros de distância, com isso o motor de acionamento para e dispara um alarme sonoro até que o obstáculo seja removido.

Os testes para definir qual a rota o protótipo iria seguir demandou mais tempo de estudo para ser concluído. Inicialmente, foi necessário inserir uma instrução para que o módulo fornecesse os pontos de localização em tempo real, exibindo-os no Serial Monitor. Desta forma, foi pesquisado um ponto fixo com latitude e longitude através do Google Maps, definido então como ponto inicial, para comparar com os dados recebidos do módulo GPS. No primeiro momento, a solução encontrada foi, fazer uma programação simples onde a intenção era acionar um led caso o ponto inicial fosse mudado para outro qualquer. A partir desse momento já era notável que seria possível colocar uma instrução para toda alteração de localização gerada pelo módulo GPS, sendo assim, foi feita um programação baseada nesse método, onde foi inserido instruções a cada mudança de direção do AVG em pontos onde seria necessário mudança de trajetória, conforme figura 35.

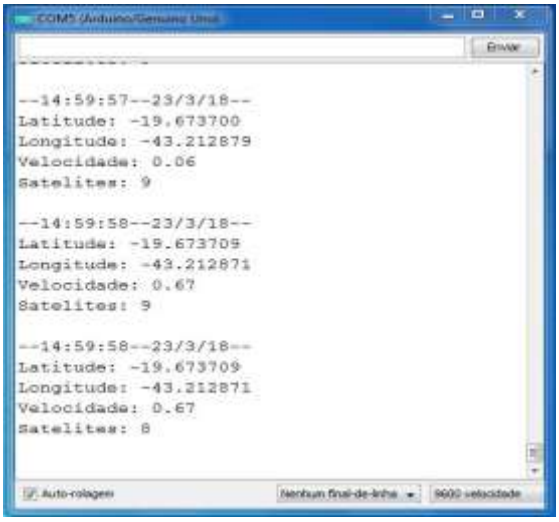

Figura 35 – *Serial Monitor* com dados de Localização
Fonte: https://portal.vidadesilicio.com.br/ (2020)

Desta forma a programação foi baseada em mapear os pontos da rota desejada inserindo as instruções para acionar o motor de direção em sentido horário fazendo que fosse possível a mudança da trajetória nos pontos predeterminados, a cada início e fim de curva existe um instrução para ativar e desativar o motor de direção, conforme figura 36.

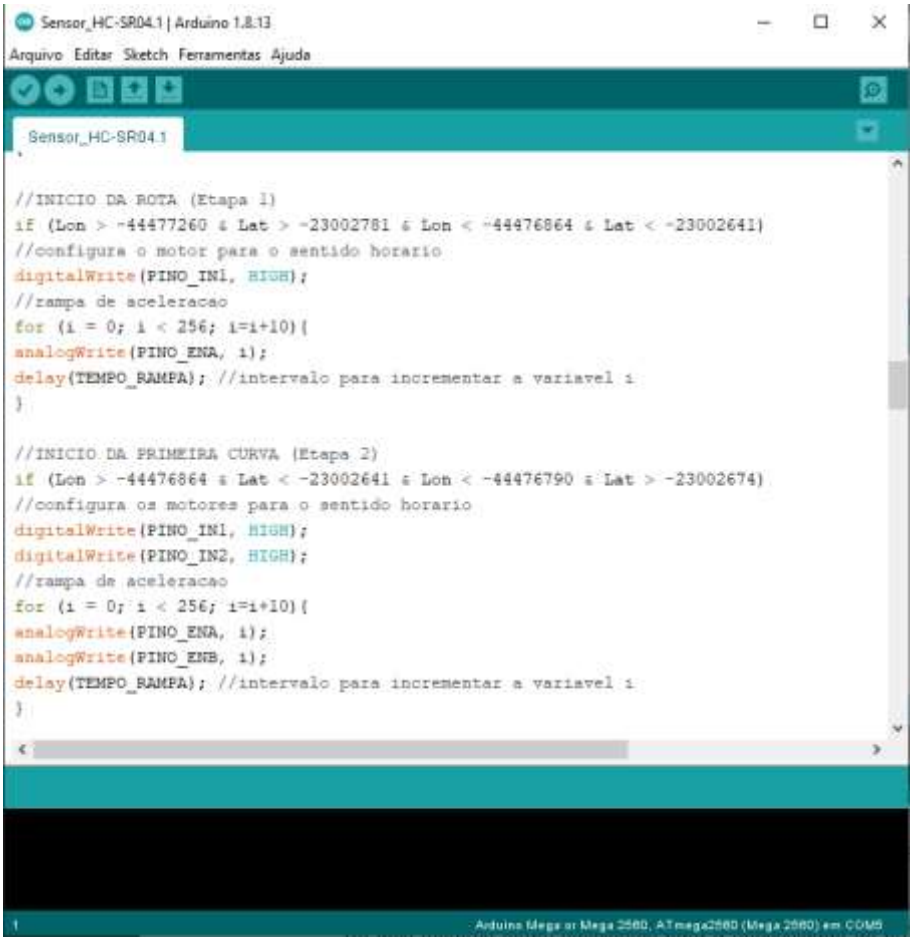

Figura 36 – Trecho da Programação do Módulo GPS
Fonte: Autor (2020)

O empenho e tempo dedicado nos estudos de algumas funções específicas do arduíno foi fundamental para o sucesso desse projeto. Todas as configurações programadas através do software IDE (Ambiante de Desenvolvimento Integrado) podem ser alteradas conforme necessário, alterar os pontos mapeados pelo GPS para modificar a trajetória do veículo, sendo possível também, modificar distância de detecção de objetos e até mesmo a configuração de acionamento dos motores, tudo isso através desse software.

5.2 Resultados Obtidos

Os testes foram realizados de forma individual no intuito de conhecer as funcionalidades de cada parte do projeto, buscando no primeiro momento o pleno funcionamento de cada um deles.

A partir do momento que todos os sistemas estavam funcionando isoladamente iniciamos a interação de todas as partes projeto.

Com o desenvolvimento do trabalho, o protótipo executa as seguintes funções:

- Seguir uma rota sinuosa pré-determinada usando as coordenadas configuradas previamente para condução do veículo.
- Segurança em parar o veículo em caso de obstáculo sem que a rota seja alterada.
- Conseguiu-se uma precisão de localização de 1,5 metros em campo aberto.
- Possível acompanhamento de todo o processo funcional enquanto conectado o protótipo em um computador por mensagens exibidas no Serial Monitor.

5.3 Problemas encontrados

Neste capítulo serão descritas as dificuldades encontradas no desenvolvimento do protótipo do AGV guiado por coordenadas geográficas.

Com o desenvolvimento do projeto em andamento, foi notório que não havia uma precisão adequada da localização do módulo GPS do Arduino, com o intuito de resolver este problema iniciamos um busca e notamos que o erro acontecia devido a quantidade de números de casas decimais das coordenadas geográficas que eram usadas na localização do AGV.

No primeiro momento, no intuito de simplificar utilizamos apenas 2 casas decimais após a identificação da longitude e latitude. Após realizarmos alguns testes notou-se que, o número de casas decimais interfere diretamente na precisão da

localização, sendo necessário utilizar 6 casas após a vírgula para conseguir uma precisão 100% segura.

No inicio do desenvolvimento do sistema de movimentação foi utilizado motores de corrente contínua e foi verificado que não seria possível liga-los diretamente nas saídas do Arduino que possuem baixo nível de corrente, sendo que, qualquer valor acima do permitido pode danifica-las de forma permanente. Foi necessário então, a utilização do “*Shield Motor Drive*” que foi conectado junto as saídas do Arduino com a função de acionar os motores de forma individual e segura sem comprometer as saídas do Arduino.

O sistema de detecção de obstáculos foi desenvolvido pensando em minimizar possíveis acidentes, entretanto, houve alguns problemas relacionados a funcionalidade do sensor. O sensor utilizado possui uma certa limitação de ângulo de operação e atraso no tempo de resposta de detecção, com isso foi necessário inserir níveis de detecção utilizando leds para sinalizar a aproximação do objeto incluindo sinalização sonora com parada do motor no último estágio de detecção.

5.3.1 Diferença da utilização das Coordenadas Geográficas

A precisão da localização está diretamente relacionada a quantidade de casas decimais utilizadas para busca de qualquer ponto na superfície terrestre, a não utilização de no mínimo 6 casas de decimais, causa a distorção representada pela figura 36, onde são utilizadas apenas 2 casas decimais, gerando uma localização diferente da prevista.

Diante disso, a fim de minizar erros na busca da localização, foi usado 6 casas decimais para as coordenadas latitude e longitude, conforme figura 37.

Segue abaixo a demostração prática da diferença de localização com o aumento da utilização das casas decimais para as seguintes coordenadas: 23°00'10.0"S 44°28'38.1"W ou -23.002781, -44.477260.

- **Usando coordenadas com 2 casas decimais.**

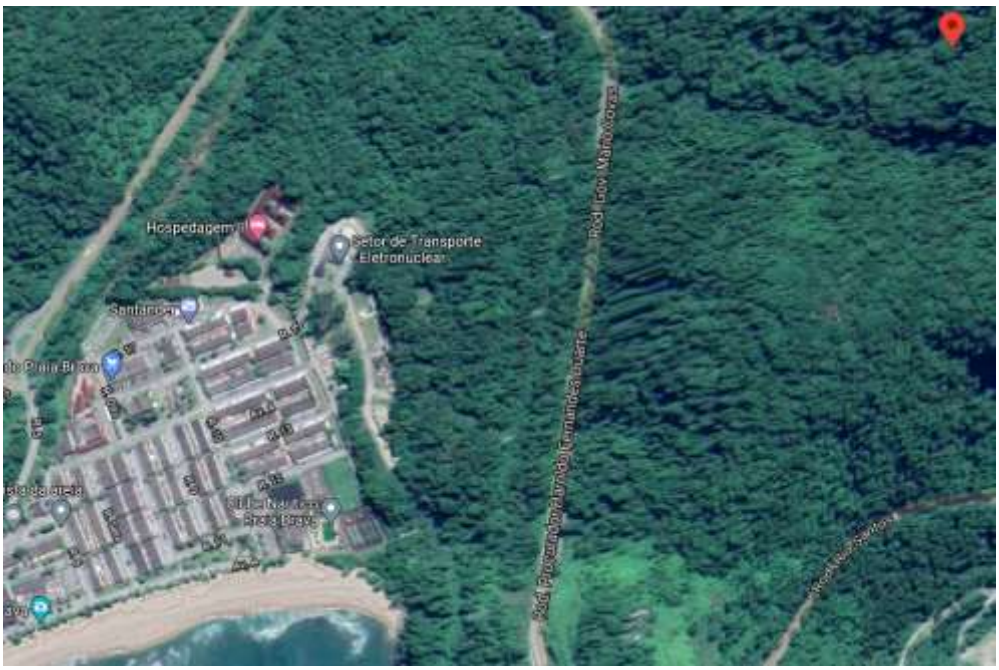

Figura 37 – Localização incorreta gerada pela utilização de duas casas decimais.
Fonte: Google Maps (2020)

- **Usando coordenadas com 6 casas decimais.**

Figura 38 – Localização correta gerada pela utilização de seis casas decimais.
Fonte: Google Maps (2020)

6 CONCLUSÃO

O mundo em que vivemos está em constante evolução de tecnologias e de globalização, fazendo com que a cada dia novas ideias e propostas surjam em vários lugares do planeta. A Engenharia, em geral, e todas as suas ramificações, tem uma parcela importante no desenvolvimento global.

A necessidade incessante de reduzir ou eliminar custos em todo o processo produtivo, incluindo diminuir a mão de obra humana em locais de alto risco, tem incentivado as empresas a buscar e investir em tecnologia de última geração, e neste sentido o uso de veículos guiados de forma autônoma tem aumentado de forma gradativa, e no cenário atual que é altamente competitivo, é de suma importância que cada vez mais as empresas busquem continuamente formas de melhorias e a implantação de novas tecnologias, o que as vezes não acontece, desta forma, os empreendimentos que não saírem da inércia poderão ter perda de mercado resultando em uma possível falência.

O presente projeto, por se tratar de um trabalho com abrangência em várias áreas específicas, possibilitou o aprendizado e aumento de conhecimento do funcionamento de dispositivos diversos.

Com o desenvolvimento desse projeto foi possível constatar a implementação de um sistema de automatismo de um veículo por meio do uso de coordenadas geográficas incluindo a aplicação de alguns sistemas, conforme descrito no capitulo 4 desta monografia.

Tendo como foco principal o desenvolvimento de um software capaz de integrar o algoritmo usando a plataforma IDE a um AGV comercial genérico, foi necessário atenção e estudo dos aspectos de comunicação, compatibilidade de hardware e integração de sistemas. A etapa de navegação, ainda não sendo capaz de apresentar um desempenho compatível com o de um produto comercial, desempenhou bem sua função e mostrou potencial para melhorias, uma vez que seu comportamento é próximo ao desejado em situações restritas, como trajetos curtos e poucas mudanças de trajetória, sendo assim, uma prova de conceito considerada satisfatória.

Portando, com base no conteúdo estudado e revisado para o desenvolvimento desta monografia, apesar de todas as dificuldades encontradas o objetivo final foi alcançado com sucesso.

7 SUGESTÕES PARA TRABALHOS FUTUROS

7.1 Sugestões para Trabalhos Futuros

Durante o processo de desenvolvimento deste projeto nos deparamos com vários obstáculos e dificuldades, e no decorrer desses 6 meses de esforço e determinação alcançamos o nos propusermos a fazer, um sistema funcional, prático e de fácil compreensão.

7.1.1 Dificuldade na precisão do Módulo GPS

Logo no início deste projeto tivemos como maior objetivo, criar um protótipo eficiente e que conseguisse fornecer as coordenadas geográficas de maneira eficiente e consistente para guiar AVG conforme o trajeto determinado.

O módulo usado possui uma pequena distorção nos dados recebidos, acontece uma oscilação no momento em que são geradas as coordenadas geográficas, fazendo com que perdêssemos nossa rota de forma inesperada.

Sendo assim, durante as pesquisas notamos que a forma mais prática seria a troca por um módulo GPS com maior precisão para que minimizar os erros de localização.

Entretanto, devido o alto custo e falta de disponibilidade, optamos por manter o módulo inicial que nos atendeu de forma satisfatória.

7.1.2 Conexão de Dados via Wi-Fi e IHM

Após a conclusão do desenvolvimento do projeto, foi possível observar que a implementação de uma conexão de rede Wi-Fi seria de extrema relevância, sendo possível gerenciar suas variáveis, alterar rota, capacidade de carga da bateria, entre outros, tudo isso através de uma IHM (Interface Homem Máquina) onde seria capaz de acompanhar todas as rotinas de funcionamento do protótipo, efetuar a troca da rota através de um lcd e comandos externo ou podendo alterar de qualquer lugar apenas conectado a essa rede, recebendo todas essas informações em tempo real usando apenas um Smartphone com sistema operacional Android.

8 REFERÊNCIAS BIBLIOGRÁFICAS

CATALAO CML. **Navegação Eletrônica**. Disponível em: <https://sites.google.com/site/catalaocml/home/nav-electronica>. Acesso em: Outubro de 2020.

CITISYSTEMS. **O Que é Indústria 4.0 e Como Ela Vai Impactar o Mundo**. Disponível em: <https://www.citisystems.com.br/industria-4-0/>. Acesso em: Maio de 2020.

Coffey, J. **Circumference of the Earth**. Disponível em: <https://www.universetoday.com /26461/ circumference-of-the-earth/>. Acesso em: Novembro de 2020

DIRECT INDUSTRY. **Como escolher um AGV**. Disponível em: <http://guide.directindustry.com/pt/que-agv-escolher/#5>. Acesso em: Maio de 2020.

ELECTRONICA PT. **Associação de Baterias**. Disponível em: <https://www.electronica-pt.com/associacao-baterias>. Acesso em: Junho de 2020.

EMBARCADOS. **Arduino comunicação serial.** Disponível em: <https://www.embarcados.com.br/arduino-comunicacao-serial>. Acesso em: Março de 2020.

EMPELTEC JR. **Arduino: Introdução A Um Universo de de Possibilidades Para Sua Automação**. Disponível em: < https://empeltecjr.com/arduino/?gclid=Cj0KCQjwz4z3BRCgARIsAES_OVcy7y6su_v68jHiGHf356> . Acesso em: Junho de 2020.

FILIPEFLOP. **Componentes Eletrônicos**. Disponível em: <www.filipeflop.com.br>. Acesso em: Setembro de 2020.

INFOESCOLA. **Pilhas e Baterias**. Disponível em: <http://www.infoescola.com/quimica/pilhas-e-baterias/>. Acesso em: Abril de 2020.

Kaplan, Elliott D. **Understanding GPS: Principles and Applications**. Editora Artech House,1996.

MASTERWALKER. **Arduino - Utilizando o Sensor Ultrassônico HC-SR04 e Buzzer 5V.** Disponível em: < https://blogmasterwalkershop.com.br/arduino/arduino-utilizando-o-sensor-ultrasonico-hcsr04-e-buzzer-5v/>. Acesso em: Agosto de 2020.

MUNDO DA ELÉTRICA. **Dicas de projetos e conceitos de eletricidade**. Disponível em: <https://www.mundodaeletrica.com.br>. Acesso em: Novembro de 2020.

MURARO, R. M. **A automação e o futuro do homem**. [S.l.]: Editora Vozes, 1969. Citado 2 vezes nas páginas 19 e 20.

MUSEU WEG DE CIÊNCIA E TECNOLOGIA. **A História do Motor Elétrico Que Você Precisa Conhcer**. Disponível em: <https://museuweg.net/blog/a-historia-do-motoreletrico/#:~:text=Levou%20quase%20tr%C3%AAs%20s%C3%A9culos%20entre,de%20corrente%20cont%C3%ADnua%20auto%20induzido>. Acesso em: Agosto de 2020.

ROBOCORE. **Arduino Shield - Motor Driver 2x2A**. Disponível em: <https://robocore.net/shields-arduino/arduino-shield-motor-driver-2x2a>. Acesso em: Novembro de 2020.

Segantine, P. C. L. **GPS: Sistema de Posicionamento Global**. Departamento de Engenharia de Transportes. EESC-USP. São Carlos (SP), 2005

SIAUT. **Sistema de Navegação**. Disponível em: <https://ave.dee.isep.ipp.pt/~mjf/act_lect/SIAUT/Trabalhos%202007-08/Trabalhos/SI AUT_Navegacao.pdf>. Acesso em: Setembro de 2020.

STA ELETRÔNICA. **Tipos de Baterias**. Disponível em: <www.staeltronica.com.br_artigos_tipos-de-baterias>. Acesso em: Junho de 2020.

TELMAC. **Motores Elétricos**. Disponível em: <http://www.telmac.com.br/motores-eletricos.html>. Acesso em: Julho de 2020.

ANEXO 1

PROGRAMAÇÃO ARDUINO

Este anexo tem como funcionalidade básica a citação completa da programação utilizada em todo o protótipo com seus devidos comentários nas respectivas linhas.

Para incluí-la é necessário que abra a insterface do Arduino com o computador e digitá-la da maneira abaixo.

```
//Inclusão das bibliotecas
#include "Ultrasonic.h"              //INCLUSÃO DA BIBLIOTECA NECESSÁRIA PARA FUNCIONAMENTO DO CÓDIGO
#include <SoftwareSerial.h>          //INCLUSÃO DA BIBLIOTECA NECESSÁRIA PARA FUNCIONAMENTO DO CÓDIGO
#include <TinyGPS.h>                 //INCLUSÃO DA BIBLIOTECA NECESSÁRIA PARA FUNCIONAMENTO DO CÓDIGO
//Declaração das Variáveis
const int echoPin = 7;               //PINO DIGITAL UTILIZADO PELO HC-SR04 ECHO(RECEBE)
const int trigPin = 8;               //PINO DIGITAL UTILIZADO PELO HC-SR04 TRIG(ENVIA)
const int Para = 2;                  //PINO DIGITAL EM QUE O LED ESTÁ CONECTADO
const int Alarme = 3;                //PINO DIGITAL EM QUE O LED ESTÁ CONECTADO
const int Sinal = 4;                 //PINO DIGITAL EM QUE O LED ESTÁ CONECTADO
const int Freio = 28;                //PINO DIGITAL EM QUE O LED ESTÁ CONECTADO
const int Esquerda = 24;             //PINO DIGITAL EM QUE O LED ESTÁ CONECTADO
const int Direita = 26;              //PINO DIGITAL EM QUE O LED ESTÁ CONECTADO
const int Frente = 22;               //PINO DIGITAL EM QUE O LED ESTÁ CONECTADO
const int PINO_ENA = 5;              //DECLARACAO DOS PINOS UTILIZADOS PARA CONTROLAR O MOTOR
const int PINO_ENB = 9;              //DECLARACAO DOS PINOS UTILIZADOS PARA CONTROLAR O MOTOR
const int PINO_IN1 = 12;             //DECLARACAO DOS PINOS UTILIZADOS PARA CONTROLAR O SENTIDO DO MOTOR
const int PINO_IN2 = 13;             //DECLARACAO DOS PINOS UTILIZADOS PARA CONTROLAR O SENTIDO DO MOTOR

SoftwareSerial serial1 (10, 11);     //INICIALIZANDO OS PINOS RX, TX
TinyGPS gps1;                        //INICIALIZANDO OS GPS
Ultrasonic ultrasonic(trigPin,echoPin);  //INICIALIZANDO OS PINOS
int distancia;                       //CRIA UMA VARIÁVEL CHAMADA "distancia" DO TIPO INTEIRO
int i = 0;                           //declaracao da variavel para as rampas
const int TEMPO_ESPERA = 1000;       //declaracao do intervalo de 1 segundo entre os sentidos de rotacao do motor
const int TEMPO_RAMPA = 30;          //declaracao do intervalo de 30 ms para as rampas de aceleracao e desaceleracao
void setup(){
pinMode(echoPin, INPUT);             //DEFINE O PINO COMO ENTRADA (RECEBE)
pinMode(trigPin, OUTPUT);            //DEFINE O PINO COMO SAÍDA (ENVIA)
pinMode(Para, OUTPUT);               //DECLARA O PINO COMO SENDO SAÍDA
pinMode(Alarme, OUTPUT);             //DECLARA O PINO COMO SENDO SAÍDA
```

```
pinMode(Sinal, OUTPUT);                 //DECLARA O PINO COMO SENDO SAÍDA
pinMode(Freio, OUTPUT);                 //DECLARA O PINO COMO SENDO SAÍDA
pinMode(Esquerda, OUTPUT);              //DECLARA O PINO COMO SENDO SAÍDA
pinMode(Direita, OUTPUT);               //DECLARA O PINO COMO SENDO SAÍDA
pinMode(Frente, OUTPUT);                //DECLARA O PINO COMO SENDO SAÍDA
pinMode(PINO_ENA, OUTPUT);              //DECLARA O PINO COMO SENDO SAÍDA
pinMode(PINO_ENB, OUTPUT);              //DECLARA O PINO COMO SENDO SAÍDA
pinMode(PINO_IN1, OUTPUT);              //DECLARA O PINO COMO SENDO SAÍDA
pinMode(PINO_IN2, OUTPUT);              //DECLARA O PINO COMO SENDO SAÍDA
//Inicia o codigo com os motores parados
digitalWrite(PINO_IN1, LOW);
digitalWrite(PINO_IN2, LOW);
digitalWrite(PINO_ENA, LOW);
digitalWrite(PINO_ENB, LOW);
// serial1.begin(9600);
Serial.begin(9600);
Serial.println("TCC AGV GPS");
Serial.println("Lendo dados do sensor...");
}
void loop(){
bool recebido = false;
while (serial1.available()) {
char cIn = serial1.read();
recebido = gps1.encode(cIn);
}
if (recebido){
Serial.println("-----------------------------------------------------------------------------------------------");
//Latitude e Longitude
long latitude, longitude;
unsigned long idadeInfo;
gps1.get_position(&latitude, &longitude, &idadeInfo);
if (latitude != TinyGPS::GPS_INVALID_F_ANGLE) {
Serial.print("Latitude: ");
Serial.println(float(latitude) / 100000, 6);
}
if (longitude != TinyGPS::GPS_INVALID_F_ANGLE) {
Serial.print("Longitude: ");
Serial.println(float(longitude) / 100000, 6);
```

```
}
long Lon=(float(longitude)* 10);
Serial.println(Lon);
long Lat=(float(latitude)* 10);
Serial.println(Lat);
{
//INICIO DA ROTA (Etapa 1)
if (Lon > -44477260 & Lat > -23002781 & Lon < -44476864 & Lat < -23002641)
//configura o motor para o sentido horario
digitalWrite(PINO_IN1, HIGH);
//Rampa de aceleracao
for (i = 0; i < 256; i=i+10){
analogWrite(PINO_ENA, i);
delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
}
//INICIO DA PRIMEIRA CURVA (Etapa 2)
if (Lon > -44476864 & Lat < -23002641 & Lon < -44476790 & Lat > -23002674)
//Configura os motores para o sentido horario
digitalWrite(PINO_IN1, HIGH);
digitalWrite(PINO_IN2, HIGH);
//Rampa de aceleracao
for (i = 0; i < 256; i=i+10){
analogWrite(PINO_ENA, i);
analogWrite(PINO_ENB, i);
delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
}
//FIM DA PRIMEIRA CURVA (Etapa 3)
if (Lon > -44476790 & Lat < -23002674 & Lon < -44476637 & Lat > -23003013)
//Configura os motores para o sentido horario
digitalWrite(PINO_IN1, HIGH);
//Rampa de desaceleracao
for (i = 255; i >= 0; i=i-10){
analogWrite(PINO_ENB, i);
delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
}
//Rampa de aceleracao
for (i = 0; i < 256; i=i+10){
```

```
analogWrite(PINO_ENA, i);
delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
}
//INICIO DA SEGUNDA CURVA (Etapa 4)
if (Lon < -44476637 & Lat < -23003013 & Lon > -44476654 & Lat > -23003068)
//configura os motores para o sentido horario
digitalWrite(PINO_IN1, HIGH);
digitalWrite(PINO_IN2, HIGH);
//Rampa de aceleracao
for (i = 0; i < 256; i=i+10){
analogWrite(PINO_ENA, i);
analogWrite(PINO_ENB, i);
delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
}
//FIM DA SEGUNDA CURVA (Etapa 5)
if (Lon < -44476654 & Lat < -23003068 & Lon > -44477842 & Lat > -23003516)
//Configura os motores para o sentido horario
digitalWrite(PINO_IN1, HIGH);
//Rampa de desaceleracao
for (i = 255; i >= 0; i=i-10){
analogWrite(PINO_ENB, i);
delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
}
//Rampa de aceleracao
for (i = 0; i < 256; i=i+10){
analogWrite(PINO_ENA, i);
delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
}
//INICIO DA TERCEIRA CURVA (Etapa 6)
if (Lon < -44477842 & Lat > -23003516 & Lon > -44477900 & Lat < -23003494)
//Configura os motores para o sentido horario
digitalWrite(PINO_IN1, HIGH);
digitalWrite(PINO_IN2, HIGH);
//Rampa de aceleracao
for (i = 0; i < 256; i=i+10){
analogWrite(PINO_ENA, i);
analogWrite(PINO_ENB, i);
delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
```

```
}
//FIM DA TERCEIRA CURVA (Etapa 7)
if (Lon < -44477900 & Lat > -23003494 & Lon > -44478060 & Lat < -23003126)
//Configura os motores para o sentido horario
digitalWrite(PINO_IN1, HIGH);
//Rampa de desaceleracao
for (i = 255; i >= 0; i=i-10){
analogWrite(PINO_ENB, i);
delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
}
//Rampa de aceleracao
for (i = 0; i < 256; i=i+10){
analogWrite(PINO_ENA, i);
delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
}
//INICIO DA QUARTA CURVA (Etapa 8)
if (Lon > -44478060 & Lat > -23003126 & Lon < -44478033 & Lat < -23003073)
//Configura os motores para o sentido horario
digitalWrite(PINO_IN1, HIGH);
digitalWrite(PINO_IN2, HIGH);
//Rampa de aceleracao
for (i = 0; i < 256; i=i+10){
analogWrite(PINO_ENA, i);
analogWrite(PINO_ENB, i);
delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
}
//FIM DA QUARTA CURVA (Etapa 9)
if (Lon > -44478033 & Lat > -23003073 & Lon < -44477270 & Lat < -23002791)
//Configura os motores para o sentido horario
digitalWrite(PINO_IN1, HIGH);
//Rampa de desaceleracao
for (i = 255; i >= 0; i=i-10){
analogWrite(PINO_ENB, i);
delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
}
//Rampa de aceleracao
for (i = 0; i < 256; i=i+10){
analogWrite(PINO_ENA, i);
```

```
delay(TEMPO_RAMPA); //intervalo para incrementar a variavel i
}
//FIM DA ROTA (FIM)
if (Lon > -44477270 & Lat > -23002791 & Lon < -44477260 & Lat < -23002781)
//Configura o motor para o sentido horario
digitalWrite(PINO_IN1, HIGH);
//Rampa de desaceleracao
for (i = 255; i >= 0; i=i-10){
analogWrite(PINO_ENA, i);
delay(TEMPO_RAMPA);
}
hcsr04(); // FAZ A CHAMADA DO MÉTODO "hcsr04()"
}}
if(distancia <= 10){              // SE A DISTÂNCIA ENTRE O OBJETO E O SENSOR ULTRASONICO FOR MENOR QUE 10CM, FAZ
digitalWrite(Para,HIGH);       //ACIONA O LED
digitalWrite(Alarme,HIGH);     //ACIONA O LED
digitalWrite(Sinal,HIGH);      //ACIONA O LED
//rampa de desaceleracao
for (i = 255; i >= 0; i=i-10){
analogWrite(PINO_ENA, i);
delay(TEMPO_RAMPA);       //intervalo para incrementar a variavel i
}
}else{//SENÃO, FAZ
digitalWrite(Para,LOW);        //LED PERMANECE DESLIGADO
if(distancia <= 20){              // SE A DISTÂNCIA ENTRE O OBJETO E O SENSOR ULTRASONICO FOR MENOR QUE 20CM, FAZ
digitalWrite(Alarme,HIGH);     //ACIONA O LED
digitalWrite(Sinal,HIGH);      //ACIONA O LED
}else{                            //SENÃO, FAZ
digitalWrite(Alarme,LOW);      //LED PERMANECE DESLIGADO
if(distancia <= 30){              // SE A DISTÂNCIA ENTRE O OBJETO E O SENSOR ULTRASONICO FOR MENOR QUE 30CM, FAZ
digitalWrite(Sinal,HIGH);      //ACIONA O LED
}else{                            //SENÃO, FAZ
digitalWrite(Sinal,LOW);       //LED PERMANECE DESLIGADO
}}}}
//MÉTODO RESPONSÁVEL POR CALCULAR A DISTÂNCIA
```

```
void hcsr04(){
digitalWrite(trigPin, LOW);              //SETA O PINO 8 COM UM PULSO BAIXO "LOW"
delayMicroseconds(1);                    // DELAY DE 1 MICROSSEGUNDO
digitalWrite(trigPin, HIGH);             //SETA O PINO 8 COM PULSO ALTO "HIGH"
delayMicroseconds(1);                    // DELAY DE 1 MICROSSEGUNDO
digitalWrite(trigPin, LOW);              //SETA O PINO 8 COM PULSO BAIXO "LOW" NOVAMENTE
//FUNÇÃO RANGING, FAZ A CONVERSÃO DO TEMPO DE
//RESPOSTA DO ECHO EM CENTÍMETROS E ARMAZENA
//NA VARIÁVEL "distancia"
distancia = (ultrasonic.Ranging(CM));  // VARIÁVEL GLOBAL RECEBE O VALOR DA DISTÂNCIA MEDIDA
delay(500);                              //INTERVALO DE 100 MILISSEGUNDOS
Serial.print(distancia);                 //IMPRIMI O VALOR DA VARIAVEL DISTANCIA
Serial.println("cm");
delay(100);
}
```

ANEXO 2

BIBLIOTECA TINYGPS.h

```
/*
TinyGPS - a small GPS library for Arduino providing basic NMEA parsing
Based on work by and "distance_to" and "course_to" courtesy of Maarten Lamers.
Suggestion to add satellites(), course_to(), and cardinal(), by Matt Monson.
Copyright (C) 2008-2012 Mikal Hart
All rights reserved.

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
*/

#ifndef TinyGPS_h
#define TinyGPS_h

#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#endif

#define _GPS_VERSION 12 // software version of this library
#define _GPS_MPH_PER_KNOT 1.15077945
#define _GPS_MPS_PER_KNOT 0.51444444
#define _GPS_KMPH_PER_KNOT 1.852
#define _GPS_MILES_PER_METER 0.00062137112
#define _GPS_KM_PER_METER 0.001
// #define _GPS_NO_STATS

class TinyGPS
{
public:
  enum {
    GPS_INVALID_AGE = 0xFFFFFFFF,      GPS_INVALID_ANGLE = 999999999,
    GPS_INVALID_ALTITUDE = 999999999,  GPS_INVALID_DATE = 0,
    GPS_INVALID_TIME = 0xFFFFFFFF,                 GPS_INVALID_SPEED = 999999999,
    GPS_INVALID_FIX_TIME = 0xFFFFFFFF, GPS_INVALID_SATELLITES = 0xFF,
    GPS_INVALID_HDOP = 0xFFFFFFFF
  };

  static const float GPS_INVALID_F_ANGLE, GPS_INVALID_F_ALTITUDE, GPS_INVALID_F_SPEED;

  TinyGPS();
  bool encode(char c); // process one character received from GPS
  TinyGPS &operator << (char c) {encode(c); return *this;}

  // lat/long in hundred thousandths of a degree and age of fix in milliseconds
  void get_position(long *latitude, long *longitude, unsigned long *fix_age = 0);

  // date as ddmmyy, time as hhmmsscc, and age in milliseconds
  void get_datetime(unsigned long *date, unsigned long *time, unsigned long *age = 0);

  // signed altitude in centimeters (from GPGGA sentence)
  inline long altitude() { return _altitude; }

  // course in last full GPRMC sentence in 100th of a degree
  inline unsigned long course() { return _course; }

  // speed in last full GPRMC sentence in 100ths of a knot
  inline unsigned long speed() { return _speed; }

  // satellites used in last full GPGGA sentence
  inline unsigned short satellites() { return _numsats; }

  // horizontal dilution of precision in 100ths
```

```
  inline unsigned long hdop() { return _hdop; }

  void f_get_position(float *latitude, float *longitude, unsigned long *fix_age = 0);
  void crack_datetime(int *year, byte *month, byte *day,
    byte *hour, byte *minute, byte *second, byte *hundredths = 0, unsigned long *fix_age = 0);
  float f_altitude();
  float f_course();
  float f_speed_knots();
  float f_speed_mph();
  float f_speed_mps();
  float f_speed_kmph();

  static int library_version() { return _GPS_VERSION; }

  static float distance_between (float lat1, float long1, float lat2, float long2);
  static float course_to (float lat1, float long1, float lat2, float long2);
  static const char *cardinal(float course);

#ifndef _GPS_NO_STATS
  void stats(unsigned long *chars, unsigned short *good_sentences, unsigned short *failed_cs);
#endif

private:
  enum {_GPS_SENTENCE_GPGGA, _GPS_SENTENCE_GPRMC, _GPS_SENTENCE_OTHER};

  // properties
  unsigned long _time, _new_time;
  unsigned long _date, _new_date;
  long _latitude, _new_latitude;
  long _longitude, _new_longitude;
  long _altitude, _new_altitude;
  unsigned long  _speed, _new_speed;
  unsigned long  _course, _new_course;
  unsigned long  _hdop, _new_hdop;
  unsigned short _numsats, _new_numsats;

  unsigned long _last_time_fix, _new_time_fix;
  unsigned long _last_position_fix, _new_position_fix;

  // parsing state variables
  byte _parity;
  bool _is_checksum_term;
  char _term[15];
  byte _sentence_type;
  byte _term_number;
  byte _term_offset;
  bool _gps_data_good;

#ifndef _GPS_NO_STATS
  // statistics
  unsigned long _encoded_characters;
  unsigned short _good_sentences;
  unsigned short _failed_checksum;
  unsigned short _passed_checksum;
#endif

  // internal utilities
  int from_hex(char a);
  unsigned long parse_decimal();
  unsigned long parse_degrees();
  bool term_complete();
  bool gpsisdigit(char c) { return c >= '0' && c <= '9'; }
  long gpsatol(const char *str);
  int gpsstrcmp(const char *str1, const char *str2);
};

#if !defined(ARDUINO)
// Arduino 0012 workaround
#undef int
#undef char
#undef long
#undef byte
#undef float
#undef abs
#undef round
```

```
#endif
#endif
```