**TURMA – CC1P06 CIÊNCIA DA COMPUTAÇÃO**

**DIEGO BOKOR DOS SANTOS RA - N291656**

**GABRIEL VICTOR TELES GUALBERTO DA SILVA RA – G86HGJ0**

**KAIO VIANA DE ARAÚJO CAMPOS RA – F3529F5**

**MATHEUS MESQUITA MATA RA – G868374**

**VITORIA APARECIDA ALMEIDA DE OLIVEIRA RA – G415HH0**

**DESENVOLVIMENTO E AUTOMAÇÃO**

**PARA PROTOTIPAGEM**

**ALPHAVILLE / SP**

**2023**

**TURMA – CC1P06 CIÊNCIA DA COMPUTAÇÃO**

**DIEGO BOKOR DOS SANTOS RA - N291656**

**GABRIEL VICTOR TELES GUALBERTO DA SILVA RA – G86HGJ0**

**KAIO VIANA DE ARAÚJO CAMPOS RA – F3529F5**

**MATHEUS MESQUITA MATA RA – G868374**

**VITORIA APARECIDA ALMEIDA DE OLIVEIRA RA – G415HH0**

**DESENVOLVIMENTO E AUTOMAÇÃO**

**PARA PROTOTIPAGEM**

**Projeto interdisciplinar apresentado ao Curso de Ciência da Computação da Universidade Paulista de São Paulo, Campus Alphaville, para disciplina de Atividades Práticas Supervisionadas.**

**ALPHAVILLE / SP**

**2023**

**Sumário**

[1 Introdução 4](#_Toc150872519)

[2 Objetivo do trabalho 5](#_Toc150872520)

[3 Dissertação Interdisciplinaridade 6](#_Toc150872521)

[4 Consciência ambiental: lixo tóxico 7](#_Toc150872522)

[5 Arduino (conceitos gerais) 9](#_Toc150872523)

[6 Conceitos detalhados sobre o Arduino 10](#_Toc150872524)

[6.1 Tipos de Arduino 10](#_Toc150872525)

[7 Benefícios em relação às outras tecnologias 12](#_Toc150872526)

[7.1 Custo acessível: 12](#_Toc150872527)

[7.2 Comunidade e suporte: 12](#_Toc150872528)

[7.3 Versatilidade: 13](#_Toc150872529)

[7.4 Componentes e acessórios disponíveis: 13](#_Toc150872530)

[7.5 Outras tecnologias: 13](#_Toc150872531)

[7.6 Plataforma aberta: 13](#_Toc150872532)

[8 Descrição dos componentes do Veículo Autônomo 14](#_Toc150872533)

[9 Projeto (estrutura das ligações) 15](#_Toc150872534)

[10 Montagem 19](#_Toc150872535)

[11 Melhorias propostas e/ou implementadas 20](#_Toc150872536)

[12 Relatório com as linhas de código do programa 21](#_Toc150872537)

[12.1 Explicação do código 25](#_Toc150872538)

[13 Referências bibliográficas 30](#_Toc150872539)

[14 Ficha de Atividades Práticas Supervisionadas 31](#_Toc150872540)

# Introdução

A elaboração e construção de um veículo autônomo envolvem um processo intrincado e diversificado, repleto de considerações ligadas ao meio ambiente e à segurança. No âmbito do desenvolvimento desse veículo, exige-se a meticulosa escolha de materiais, destacando aqueles cujo descarte inadequado pode resultar em danos significativos ao meio ambiente e à saúde pública.

Essa seleção minuciosa de materiais não tem apenas em mira o desempenho e a eficiência do veículo, mas também a minimização de quaisquer impactos negativos que possam surgir ao longo de sua vida útil. Isso abrange o compromisso de assegurar que os materiais empregados sejam gerenciados de forma apropriada em todas as fases do ciclo de vida do veículo, abrangendo desde a produção até a disposição final.

Assim sendo, a criação e implementação de veículos autônomos que se atentem não somente à sua operação, mas também às implicações ambientais e à segurança pública, representam um notável avanço na busca por soluções sustentáveis e seguras no campo da mobilidade, ao mesmo tempo em que contribuem para reduzir possíveis danos tanto ao meio ambiente quanto às populações humanas e animais.

Neste trabalho será mostrado os passos da concepção do veículo desde seu início até seu modelo final, suas características e materiais utilizados, para seguir um determinado percurso em seu objetivo final, alcancem o local para a retirada adequada do material tóxico.

# Objetivo do trabalho

O processo de concepção e construção de um veículo autônomo é um empreendimento complexo e multifacetado que se depara com uma série de considerações tecnológicas e de segurança. No âmbito do desenvolvimento de tal veículo, impõe-se uma restrita seleção de materiais, com ênfase naquele cujo descarte inadequado pode acarretar impactos prejudiciais ao meio ambiente e à saúde pública.

Essa escolha criteriosa de materiais visa não apenas a eficiência e desempenho do veículo, mas também a minimização do potencial impacto negativo que pode advir de seu ciclo de vida. Isso inclui o compromisso de garantir que os materiais utilizados sejam gerenciados de maneira adequada durante todo o ciclo de vida do veículo, desde a produção até o seu descarte.

Como resultado final do projeto, teremos um veículo ágil que tenha a capacidade de seguir determinados trajetos, que será utilizado para retirada de materiais tóxicos em determinada área restrita.

# Dissertação Interdisciplinaridade

# Consciência ambiental: lixo tóxico

A consciência ambiental refere-se à**compreensão e à preocupação com os impactos das atividades humanas no meio ambiente** e à adoção de práticas sustentáveis para preservar e proteger os recursos naturais. Trata-se de reconhecer a interdependência entre os seres humanos e o meio ambiente, bem como a necessidade de **promover um equilíbrio** entre o desenvolvimento socioeconômico e a preservação ambiental.

A consciência ambiental ganhou destaque a partir da segunda metade do século XX, quando a **preocupação com a degradação ambiental e seus efeitos nocivos** se intensificou. A emergência de problemas como a poluição do ar e da água, a destruição de habitats naturais, o esgotamento dos recursos naturais e as mudanças climáticas levaram a um despertar global sobre a importância da conservação ambiental.

Os resíduos tóxicos são aqueles que apresentam algum grau de toxicidade e oferecem riscos ao meio ambiente e à saúde. Assim, o lixo tóxico é todo o resíduo descartável que contém [compostos químicos](https://www.infoescola.com/compostos-quimicos/) e pode ser gerado por empresas, indústrias, hospitais, laboratórios, institutos de pesquisa e residências. Esses componentes podem contaminar o solo, a água e o ar, podendo entrar no organismo através do contato direto, ou até mesmo pela respiração. São exemplos de resíduos tóxicos as baterias de carro, pilhas, lâmpadas e equipamentos eletrônicos. Por isso é necessário o descarte adequado desses resíduos, evitando comprometer a qualidade de vida do ser humano e também o meio ambiente.

Na Constituição Federal de 1988, o Artigo 225 do Meio Ambiente aponta que a empresa que comercializa produtos classificados como tóxicos deve recolher e destinar os mesmos quando estes não estão em uso, demonstrando uma preocupação com o meio ambiente e os [seres vivos](https://www.infoescola.com/biologia/os-seres-vivos/). As empresas detêm diferentes meios de recolher esse material, como contratar pessoas responsáveis pela retirada, ou até oferecer desconto em novos produtos em troca do velho como é o caso de aparelhos celulares. Além disso, boa parte dos estabelecimentos possui balcões de atendimento onde o cliente pode descartar material tóxico como pilhas e baterias. Dessa forma é possível entrar em contato com as empresas para retirada de produtos como bateria de celular, aparelhos eletrônicos danificados e pilhas, a fim de serem destinados corretamente. Essa destinação não inclui aterros sanitários ou lixões.

O lixo tóxico pode causar diversos danos à saúde e ao meio ambiente, e constantemente são descobertos novos problemas. Um problema bastante frequente em relação a esses resíduos é a [intoxicação](https://www.infoescola.com/saude/intoxicacao/) através de seus componentes químicos, causado em sua maioria por [pilhas e baterias](https://www.infoescola.com/quimica/pilhas-e-baterias/) velhas. Em casos mais graves, podem se desenvolver tumores malignos nos ovários e nas mamas no caso das mulheres, e nos [testículos](https://www.infoescola.com/sistema-reprodutor/testiculos/) e [próstata](https://www.infoescola.com/sistema-reprodutor/prostata/) no caso dos homens. Apesar destes serem os tumores mais comuns relacionados à resíduos tóxicos, podem existir casos em que outras áreas são afetadas.

Apesar desse tipo de resíduo ser gerado em maior quantidade por indústrias de diferentes segmentos, também é possível encontrar nas residências diferentes tipos de produtos com potencial de contaminação ambiental, devido sua composição química. Por este motivo é importante que se tenha cuidado com as formas de armazenamento, preservação e destinação principalmente de celulares, pilhas, baterias e computadores. Dessa forma, cuidamos do meio ambiente e da saúde, prevenindo doenças e contaminações que afetam a qualidade de vida e até mesmo [recursos naturais](https://www.infoescola.com/ecologia/recursos-naturais/).

# Arduino (conceitos gerais)

O Arduino é uma plataforma de criação de projetos e protótipos eletrônicos, sua placa é única e de código aberto, é um projeto que engloba software e hardware que possibilita a qualquer pessoa ter acesso e realizar seus projetos utilizando um microcontrolador.

Faz parte do meio da computação física, onde o software interage diretamente com o hardware, isso torna a interação fácil com o uso de sensores, motores e diversos outros dispositivos.

Uma placa que cabe na palma da mão, é um computador como qualquer outro, possui microprocessador, memória RAM, memória flash, temporizadores dentre outras funcionalidades.

O projeto mais utilizado está na versão Uno R3, que conta com as seguintes configurações; clock de 16Mhz, 2kB de memória RAM, 32kB de memória flash, 20 portas digitais e 6 entradas analógicas.

As diferenças entre um computador de mesa e o Arduino é que, o Arduino tem o porte menor, tanto de tamanho quanto de processamento, utiliza diferentes entradas e saídas em geral, com o Arduino estamos mais perto do meio físico, podemos utilizar e ler dados de sensores, controlar outros circuitos elétricos (lâmpadas, eletrodomésticos), coisas que dificilmente conseguiremos utilizando um PC de mesa.

Nós mesmos podemos construir os circuitos que usamos, porém estamos limitados ao uso de componentes existentes no mercado.

# Conceitos detalhados sobre o Arduino

O Arduino é uma plataforma de desenvolvimento de hardware de código aberto amplamente utilizada para prototipagem rápida e projetos eletrônicos. Seus conceitos básicos incluem:

Microcontrolador: O Arduino é baseado em um microcontrolador, como o ATmega328 no Arduino Uno, que executa o código programado e controla os componentes eletrônicos.

IDE (Ambiente de Desenvolvimento Integrado): O Arduino IDE é uma ferramenta de software usada para escrever, compilar e carregar código para o Arduino. É uma interface amigável para programadores iniciantes.

Linguagem de Programação: O Arduino utiliza uma linguagem de programação baseada em C/C++, simplificada para facilitar o uso. Os programas são chamados de "sketches" e consistem nas funções setup() e loop().

Pinos de E/S: Os Arduinos possuem pinos de entrada/saída digital (I/O) e analógicos que podem ser usados para se comunicar com sensores, atuadores e outros dispositivos externos.

Comunicação Serial: Os Arduinos podem se comunicar com outros dispositivos e computadores por meio de portas seriais, USB ou comunicação sem fio, como Bluetooth ou Wi-Fi.

Bibliotecas: O Arduino possui uma vasta coleção de bibliotecas pré-desenvolvidas que simplificam a programação de tarefas comuns, como controle de motores, exibição de informações em displays.

## Tipos de Arduino

Arduino Uno: É um dos modelos mais populares e é ideal para iniciantes. Ele possui 14 pinos digitais, 6 pinos analógicos e é baseado no microcontrolador ATmega328P.

Arduino Mega: É uma versão mais avançada com mais pinos de E/S, o que é útil para projetos maiores e mais complexos. Ele é baseado em microcontroladores ATmega2560.

Arduino Nano: É uma versão compacta do Uno, adequada para projetos com espaço limitado. Ele tem recursos semelhantes ao Uno.

Arduino Due: Utiliza um microcontrolador ARM Cortex-M3 mais poderoso, tornando-o adequado para projetos que exigem mais processamento e recursos.

Arduino Leonardo: É notável por usar o microcontrolador ATmega32u4, que suporta emulação de dispositivos HID (Human Interface Device), como teclados e mouses.

Arduino Série MKR: A série MKR consiste em placas compactas e poderosas projetadas para projetos de IoT (Internet das Coisas).

Arduino Zero: Utiliza o microcontrolador SAMD21, baseado na arquitetura ARM, oferecendo desempenho e eficiência.

# Benefícios em relação às outras tecnologias

O Arduino é conhecido por sua facilidade de uso, especialmente para iniciantes. A plataforma possui uma linguagem de programação simples baseada em C/C++, uma ampla variedade de bibliotecas disponíveis e uma comunidade ativa de desenvolvedores que compartilham projetos e soluções.

Outras tecnologias: Muitas outras tecnologias requerem conhecimento técnico avançado em eletrônica e programação, tornando-as menos acessíveis para iniciantes como, por exemplo, Raspberry, Microcontroladores PIC, a plataforma Galileo.

## Custo acessível:

As placas Arduino são relativamente baratas e com bastante disponibilidade, tornando-as acessíveis para estudantes, programadores mais experientes e até mesmo para aqueles que nunca ouviram falar da plataforma

Outras tecnologias como Particle Photon, Raspberry e o Circuit Python são alternativas mais caras, ainda mais junto ao uso de microcontroladores de marcas específicas ou kits de desenvolvimento, podem ser passar do dobro do valor que se usaria ao aderir o uso do arduino

## Comunidade e suporte:

O Arduino possui uma comunidade global ativa que oferece suporte técnico, compartilha projetos e fornece documentação detalhada, códigos para diversas funcionalidades e projetos, além de se ajudarem em fóruns, grupos em Discord e redes sociais. Muitas outras tecnologias podem carecer de uma comunidade tão unida e acessível, o que pode dificultar a resolução de problemas e o aprendizado.

## Versatilidade:

O Arduino é altamente versátil e pode ser usados em uma ampla gama de aplicações, desde projetos simples até sistemas complexos de automação como portas, sensores de temperatura e luz, sistemas de segurança pequenos, fechaduras eletrônicas.

Outras tecnologias: Algumas tecnologias são mais especializadas e têm aplicações limitadas ao uso correto, algumas não são possíveis usá-las em conjunto a outras tecnologias.

## Componentes e acessórios disponíveis:

Há uma grande variedade de módulos e acessórios disponíveis para o Arduino, como sensores, displays, motores e módulos de comunicação, que são fáceis de achar e fazer a compatibilidade.

## Outras tecnologias:

Embora outras tecnologias também tenham componentes disponíveis, a variedade e acessibilidade do ecossistema Arduino são notáveis.

## Plataforma aberta:

O Arduino é uma plataforma de código aberto, o que significa que os esquemas e o código-fonte estão disponíveis para o público. Isso promove a inovação e permite que os usuários personalizem e adaptem a plataforma de acordo com suas necessidades.

Outras tecnologias: Algumas tecnologias podem ser proprietárias e limitar a capacidade dos usuários de modificar ou personalizar o hardware e o software.

Em resumo, o Arduino oferece benefícios significativos em termos de facilidade de uso, acessibilidade, suporte da comunidade e versatilidade em comparação com outras tecnologias. É uma escolha popular para projetos de eletrônica e automação, tanto para iniciantes quanto para desenvolvedores experientes.

# Descrição dos componentes do Veículo Autônomo

Arduino: O Arduino é uma plataforma de prototipagem eletrônica com um microcontrolador programável. Ele atua como o cérebro do veículo, controlando todos os outros componentes.

Sensor Ultrassônico: Utilizado para medir a distância entre o veículo e obstáculos próximos, permitindo evitar colisões.

Sensor de Linha: Detecta linhas no chão, permitindo que o veículo siga um trajeto pré-determinado.

Motores: Responsáveis por mover o veículo. Podem ser controlados pelo Arduino para permitir movimentos para frente, para trás e curvas.

Bateria: Fornece energia para o funcionamento do veículo autônomo. Uma bateria recarregável é comumente utilizada.

Ponte H Shield L293D Driver: Componente eletrônico utilizado para controlar motores DC. Permite o controle da velocidade e direção de rotação dos motores, sendo facilmente integrado com o Arduino. Possui proteção contra sobrecorrente térmica e curto-circuito.

LEDs: Os LEDs são componentes eletrônicos que emitem luz quando uma corrente elétrica passa por eles. Podem ser utilizados para indicar o status do veículo autônomo, como ligado/desligado, detecção de obstáculos, entre outros.

Interruptor: O interruptor é um componente que permite ligar ou desligar o veículo autônomo. Pode ser utilizado como um botão de controle para iniciar ou interromper o funcionamento do veículo.

# Projeto (estrutura das ligações)

O projeto que desenvolvemos utilizando o Arduino UNO R3, condiz em um carrinho autônomo seguidor de linha branca em uma pista de cor preta, com sensores infravermelhos para identificar a mudança de cores através da luz refletida e sensor de distância para identificar o obstáculo à frente.

Quando ligado através do interruptor, irá aguardar 5 segundos e acender um led verde, em seguida o carrinho irá andar de acordo com a linha branca, caso saia da linha, irá se ajustar automaticamente e caso tenha um obstáculo a frente, irá parar em uma distância de menor ou igual a 5cm, acender um led vermelho, aguardar 5 segundos e depois desviar do obstáculo. Isso até chegar ao final da pista. Desenvolvemos um fluxograma para entendimento melhor da lógica aplicada na programação do carrinho.

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

O fluxograma nos ajuda a analisar todas as funções do carrinho de forma geral e organizar os comandos para então ser desenvolvido a programação na IDE e na linguagem.

Tendo a lógica realizada, desenvolvemos a programação na IDE do Arduino e montamos o esquema dos circuitos no Fritzing para melhor demonstração da explicação das estruturas.

Diagrama, Esquemático

Descrição gerada automaticamente

Ordenamos as cores dos fios de acordo com a sua funcionalidade, os fios azuis são os conectados nas portas GND, conhecido como neutro ou negativo. Já os fios vermelhos são os positivos, que estão conectados na porta 5V. As restantes das cores foram utilizadas apenas para maior facilidade na identificação sobre qual componente ela está conectada.

Utilizamos dois resistores, pois os leds não suportam 5V, ocasionando a queima dos componentes, por isso colocamos eles juntos aos leds na protoboard ou também conhecida como placa de ensaio. A protoboard nos ajuda a conectar componentes sem a necessidade da solda, note que alguns foram soldados direto na placa do Arduino, porém os leds e os resistores estão apenas alinhados na vertical para realizar a conexão. Por baixo da aparência da placa de ensaio, possui fileiras de metal para conduzir a energia, por isso ao encostar os pinos dentro dos furinhos, a energia é transmitida em todo o filamento.

A ponte H (componente vermelho), é responsável por variar ou inverter o sentido da corrente, além de controlar a sua potência, comumente utilizada para motores DC em Arduino, conforme a imagem.

Na parte de cima, temos o motor da direita, onde possui dois fios, um para o motor e outro para a direção dele. Isso serve para o debaixo, que seria o lado esquerdo. Para conectar eles no Arduino, foram colocados fios na parte lateral esquerda da ponte H nas respectivas portas 5,6,7 e 8 no Arduino.

Na ponte H, temos outras duas conexões chamadas GND e 5V, o GND está sendo conectado da bateria, para a ponte H e em seguida para o Arduino. Já o 5V sai da bateria para o interruptor, fazendo com que tenha o controle de liga e desliga do carrinho através da posição do interruptor.

Um pouco a frente, na parte de cima da protoboard e na parte debaixo, temos os sensores infravermelhos, responsáveis pela detecção da luz refletida da pista, diferenciando branco e preto. Ambos precisam ficar a frente e próximos da roda para ter uma melhor direção e não ter atraso entre os motores e os sensores. Importante esses componentes também estarem próximos ao chão para obter uma melhor leitura.

Cada um possui 3 conexões, sendo elas: o OUT, GND e VCC. O primeiro vai definir em que porta está o sensor, para os dois casos temos as portas A0 e A1. O GND, como já explicado, se refere ao neutro ou negativo, que deve ser ligado ao GND do Arduino e o VCC se refere ao fio positivo, que deve ser ligado à porta 5V no Arduino.

Mais à frente, temos o sensor de obstáculo, que com as suas funcionalidades de trigger e echo, nos diz a distância do nosso carrinho em relação ao obstáculo. Onde o trigger emite uma onda ultrassônica, bate no objeto e retorna pelo echo. Em suas conexões, temos o VCC, que é o fio positivo, o trigger ligado na porta 12 e echo na porta 11 do Arduino.

# Montagem

Em desenvolvimento

# Melhorias propostas e/ou implementadas

Em desenvolvimento

# Relatório com as linhas de código do programa

//Declaração de variáveis de acordo com as portas que estão conectadas

#define trigger 12

#define echo 11

#define motorD 5

#define motorE 6

#define dirD 7

#define dirE 8

#define Sensor1 A0

#define Sensor2 A1

#define ledVerde 13

#define ledVermelho 10

/\*Declarando as variaveis que irão armazenar valores

de acordo com a leitura dos sensores \*/

int valorSensor1 = 0;

int valorSensor2 = 0;

int duracao = 0;

int distancia = 0;

int velocidade = 150;

//Configurações iniciais

//O input significa que o valor será obtido a partir da leitura, usado para os sensores.

//O Output significa que as variaveis irão apenas retornar um valor, sem obter nada do externo.

void setup() {

pinMode(ledVermelho, OUTPUT);

pinMode(ledVerde, OUTPUT);

delay(5000);

pinMode(motorD, OUTPUT);

pinMode(motorE, OUTPUT);

pinMode(dirD, OUTPUT);

pinMode(dirE, OUTPUT);

pinMode(Sensor1, INPUT);

pinMode(Sensor2, INPUT);

pinMode(trigger, OUTPUT);

pinMode(echo, INPUT);

//Configuramos as direções dos motores como 0 ou LOW, para o carrinho seguir para frente inicialmente

digitalWrite(dirD, HIGH);

digitalWrite(dirE, HIGH);

//é usado para definir uma frequencia dos dados expostos no print (9600).

Serial.begin(9600);

}

void loop() {

//Leitura dos valores dos sensores infravermelhos

valorSensor1 = analogRead(Sensor1);

valorSensor2 = analogRead(Sensor2);

//Exibição dos dados no prompt da IDE

Serial.print("Leitura do Sensor1: ");

Serial.println(valorSensor1);

Serial.print("Leitura do Sensor2: ");

Serial.println(valorSensor2);

//O trigger emite o pulso de onda a cada 10 microssegundos

digitalWrite(trigger, HIGH);

delayMicroseconds(10);

digitalWrite(trigger, LOW);

//O echo recebe o pulso retornado e armazena da duracao

duracao = pulseIn(echo, HIGH);

//Calculo para encontrar a distância, o valor fixo é definido pelo cálculo da velocidade da onda ultrassônica

distancia = duracao\*0.017175;

/\*Os sensores infravermelhos retornam valores dependendo da luz refletida.

Se for branco, retorna menos de 900. Se for preto, retorna superior. \*/

if (distancia>5){

digitalWrite(ledVerde,HIGH);

digitalWrite(ledVermelho, LOW);

if ((valorSensor1 > 900) && (valorSensor2 > 900)){

digitalWrite(motorD, velocidade);

digitalWrite(motorE, velocidade);

}

if ((valorSensor1 < 900) && (valorSensor2 > 900)){

digitalWrite(motorD, 0);

digitalWrite(motorE, velocidade);

}

if ((valorSensor1 > 900) && (valorSensor2 < 900)){

digitalWrite(motorD, velocidade);

digitalWrite(motorE, 0);

}

}

/\*O if acima verifica se a distância é superior a 5cm.

Caso for igual ou inferior, irá executar o comando do else\*/

else {

Serial.println("Objeto detectado");

digitalWrite(ledVerde, LOW);

digitalWrite(ledVermelho, HIGH);

digitalWrite(motorD, 0);

digitalWrite(motorE, 0);

delay(5000);

digitalWrite(motorD, velocidade);

digitalWrite(motorE, 0);

delay(2000);

}

}

## Explicação do código

O código foi desenvolvido na linguagem C++, comumente utilizado para programação em Arduino. Escolhemos ela pois além de ser a mais comum na utilização, encontramos diversos conteúdos e exemplos de código nessa linguagem.

O que difere a programação em Arduino e a tradicional para computador, é que nessa primeira, no momento da declaração da variável, é necessário especificar a porta onde esse componente está conectado. Ou seja, a programação e a montagem do carrinho precisa estar condizente, caso uma porta esteja declarada diferente da que está na montagem, aquele componente não irá funcionar como deveria.

Além disso, o Arduino possui seus próprios comandos, como o digitalWrite e o AnalogWrite, ambos vão depender do tipo de componente e sua conexão na placa. O digitalWrite é utilizado para portas digitais, onde só é possível armazenar dois valores, 0 ou 1, onde neste caso se aplica o exemplo de desligado ou ligado. Podemos ver isso sendo aplicado nas variáveis dos LEDs e dos motores, onde para a manipulação foi utilizado apenas o LOW ou HIGH, para desligar ou ligar os componentes.

Já o analogWrite é utilizado para componentes onde possuem variedades de tensão, podendo ser de 0 até 1023. Como exemplo, temos os sensores infravermelhos e ultrassônico, onde foi armazenado os valores obtidos pela leitura e manipulamos a nossa maneira utilizando parâmetros nas estruturas de condições.

No caso dos sensores infravermelhos, para o carrinho poder andar somente na linha branca, foi criado um código para identificar quando ele estaria na superfície preta e na branca, nisso dependendo da situação, os dois motores estariam ativados (HIGH) ou um deles era desativado (LOW) para realizar a curva e se ajustar novamente na linha branca. Para compreender melhor, segue a imagem retirada de uma das referências bibliográficas.

Uma imagem contendo Diagrama

Descrição gerada automaticamente

Note que os dois sensores (LDR1 e LD2) estão sob a superfície branca (neste exemplo, o carrinho é para andar na linha preta, porém no nosso caso é ao contrário) para o carrinho poder seguir na linha preta. Usamos essa mesma lógica na nossa programação.

Para desenvolver nosso código, realizamos uma simulação no Tinkercad para testar a programação antes da montagem, com os prints no prompt podemos entender como é visualizado a leitura dos sensores, neste caso abaixo os dois estão sob a superfície preta, obtendo a leitura de 1022, sendo identificado pelo ícone oposto ao sol.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Já se um dos dois estiverem no ícone do Sol, o valor do sensor iria diminuir para abaixo de 900 (em alguns testes, os sensores davam 813, por isso colocamos a margem de 900 na programação)Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Como pode ser visto, esses valores não são possíveis de serem armazenados como 0 ou 1.

Outro exemplo é o sensor ultrassônico, que possui duas variáveis, o trigger e o echo. O trigger é o sinal ou pulso de onda enviado para detectar se tem algum obstáculo, o seu limite de área fica em torno dos 4 metros, e o mínimo é 2cm.

Já o echo é o sinal retornado dessa transmissão do trigger, ou seja, quando bater em um obstáculo, ele irá retornar o echo, com isso sabemos qual a distância do objeto e o sensor.

O cálculo da distância é feito a partir do tempo de duração que teve o retorno do echo multiplicado pelo valor 0.017175, determinado pelo cálculo:

(343,5 m/s \* 100 cm)/(1.000.000) = 0,3435

0,3435/2 = 0,017175

Onde 343,5m/s é a velocidade da onda ultrassônica. Já os outros valores se dão ao fato do Trigger enviar pulsos de onda a cada 10 microssegundos para retornar ao echo. Com isso, conforme o código, o cálculo da distância é:

distancia = duracao\*0.017175;

Na nossa programação, ao identificar um obstáculo com distancia de mais ou menos 5cm, onde adotamos o menor ou igual a 5cm, deve piscar um led vermelho e parar o carrinho.Uma imagem contendo Texto

Descrição gerada automaticamente

Note que o led vermelho está acesso, os motores estão com 0 de velocidade, ou seja, estão parados, e no prompt está escrito “Objeto detectado”, isso por causa da distância de 3.3cm.

Em resumo, esses são os pontos principais a serem explicados em relação ao código, o restante é determinado seguindo a lógica do carrinho:

Acender um led verde após 5 segundos ao iniciar o carrinho

Seguir a linha branca na pista

Identificar obstáculos e parar 5cm antes dele

Acender um led vermelho ao identificar o objeto

Desviar dos obstáculos

Voltar ao curso da linha branca em até 5 segundos

Finalizar o trajeto

# Referências bibliográficas

https://conceito.de/arduino

https://cba.ed.carlosdelfino.eti.br/conceitos-basicos

https://blog.betrybe.com/tecnologia/arduino-tudo-sobre/

https://sites.usp.br/ltsi/introducao/

https://www.arduino.cc/

https://forum.arduino.cc/

https://victorvision.com.br/blog/o-que-e-arduino/

https://blog.eletrogate.com/o-que-e-arduino-para-que-serve-vantagens-e-como-utilizar/

https://myrobot.com.br/voce-sabe-os-beneficios-da-aprendizagem-com-arduino/

https://editorarealize.com.br/editora/anais/conedu/2022/TRABALHO\_COMPLETO\_EV174\_MD1\_ID17596\_TB4581\_29112022235911.pdf

https://www.robocore.net/tutoriais/robo-seguidor-de-linha

https://www.makerhero.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/

https://lobodarobotica.com/blog/tinkercad-aprenda-a-simular-oarduino-online/#:~:text=Algo%20que%20se%20deve%20lembrar,em%20blocos%20ou%20blocos%2Btexto.

https://youtu.be/I0iOPzbzbuE?si=691xGcm3BhHTdHB2

https://youtu.be/61M7Mg7KP6k?si=XjPi1TqvDlyfJmj8

https://blog.eletrogate.com/robo-seguidor-de-linha-tutorial-completo/

# Ficha de Atividades Práticas Supervisionadas