**LAMBO CONTROL: CARRINHO COM CONTROLE DE DISTÂNCIA**

**Projeto Semestral – Instrumentação / Microcontroladores e Sistemas Microcontrolados / Sistemas de Controle I**

Ricardo Cabral Banci ¹ 20.95005-5

Guilherme De Gaspari Ximenes ¹ 19.00206-8

Hakel Boscolo Dias Blanco¹ 20.00403-6

Breno Yukio Harazaki ¹ 20.01303-5

¹ Aluno do Instituto Mauá de Tecnologia (IMT);

**Resumo.** *A implementação do Lambo Control como um exemplo prático de sistema de controle, destacando a integração de conceitos teóricos e práticos nas áreas de Instrumentação, Microcontroladores e Sistemas de Controle. O estudo centrou-se na modelagem do Lambo Control, um carro autônomo, utilizando componentes eletrônicos como, motor DC, sensor ultrassônico HC SR04, driver de ponte H L298N, adaptador CP2102 USB-Serial TTL e o microcontrolador PIC 16F1619, que foram cuidadosamente selecionados para garantir a eficácia do sistema. A metodologia adotada envolveu uma abordagem abrangente de modelagem do sistema, utilizando equações diferenciais. Foi implementado um controlador PID para garantir um controle preciso do sistema. No entanto, durante a implementação, constatou-se que a utilização do PD foi mais eficaz, evitando problemas de overshoot, multiplicando a saída por uma constante gerada pelo MATLAB. A implementação bem-sucedida do sistema de controle começou com uma lógica if simples para a realimentação do motor. Isso resultou em resultados satisfatórios, mantendo o carro autônomo a uma distância segura de um alvo para evitar colisões. O projeto demonstrou a viabilidade prática dos conceitos de controle de posição. A calibração cuidadosa dos componentes eletrônicos, a escolha do controlador adequado e a integração eficiente dos sistemas foram aspectos essenciais para o sucesso do projeto.*

*Palavras-chave: Lambo Control, Controle de Posição, Microcontroladores, Instrumentação, Sistema Autônomo.*

**Introdução**

É essencial para o avanço da engenharia de controle, este projeto pois propõe aprimorar o conhecimento e a eficiência de um sistema de carrinho com controle ultrassônico. Este sistema possui aplicações potenciais em diversos setores industriais e educacionais, proporcionando uma compreensão mais profunda dos princípios fundamentais do controle de posição. A concepção dele baseia-se na construção de um carrinho que, com um sensor ultrassônico mede a distância de um objeto, e com um motor, aproxima ou afasta a distância, criando um sistema de malha fechada. Essa proposta representa um casamento entre alguns campos da engenharia. Desafios inerentes ao controle de posição em um carrinho, identificando lacunas no entendimento atual e propondo soluções práticas para aprimorar a estabilidade e precisão do sistema são necessárias devido ao crescente demanda por sistemas de controle mais eficientes, a necessidade de uma abordagem técnica e metodológica inovadora se torna evidente em ambientes dinâmicos, assim envolvidos em matérias aprofundadas para o melhor entendimento do projeto.

A Instrumentação desempenha um papel fundamental na coleta precisa de dados, neste caso por meio de um sensor, o ultrassônico mencionado, capaz de medir distâncias e condições do ambiente.

Os Microcontroladores entram em cena como o cérebro dessa operação, integrando os dados provenientes da instrumentação e executando os algoritmos de controle através do PIC. Eles garantem que o veículo intérprete as informações corretamente e responda de maneira adequada, mantendo a distância pré-determinada em relação ao alvo, neste caso através do Software MPLAB 5.50.

A implementação de Sistemas de Controle é crucial para garantir a eficácia desse projeto. O Software MatLab oferece ferramentas para modelar, simular e otimizar os sistemas, permitindo a criação e aperfeiçoamento dos algoritmos que regem o comportamento autônomo do carro.

# **Material e Métodos**

**Materiais utilizados:**

**Sensor ultrassónico HC-SR04**

O Sensor Ultrassônico HC-SR04 desempenha um papel crucial no sistema Lanbo Control, oferecendo uma solução precisa e eficaz para a medição de distâncias em tempo real. Este dispositivo é amplamente reconhecido por sua confiabilidade e simplicidade de uso, tornando-se uma escolha popular em aplicações que exigem sensibilidade e precisão na detecção de objetos.

Seu princípio é o eco para medir distâncias. Ele emite pulsos ultrassônicos curtos e aguarda o retorno do eco gerado ao encontrar um objeto. A partir do tempo decorrido entre a emissão e a recepção do sinal, o sensor calcula a distância com base na velocidade do som. Esse método oferece uma medição precisa e rápida, tornando-o ideal para aplicações em que o tempo de resposta é crucial.

Características Principais:

* **Alcance:** O HC-SR04 tem um alcance operacional típico de 2 cm a 4 metros, tornando-o versátil para diversas situações.
* **Precisão:** Com uma resolução de medição de milímetros, o sensor proporciona uma precisão notável nas leituras.
* **Facilidade de Integração:** Sua interface simples, que inclui apenas pinos de alimentação, terra, e pinos de sinal, facilita a integração com o microcontrolador PIC no Lando Control.
* **Baixo Custo:** Uma característica atraente é o seu custo acessível, garantindo uma solução econômica para aplicações de controle de posição.

Microfone visto de perto

Descrição gerada automaticamente com confiança média

Figura 1 - Sensor Ultrassônico HC SR04

**Microcontrolador PIC 16F1619**

O PIC16F1619 é um microcontrolador da família PIC da Microchip, baseado em arquitetura de 8 bits Enhanced Mid-Range. Com 14 KB de memória Flash para o programa e 1 KB de RAM, oferece recursos avançados. Possui 256 bytes de EEPROM para armazenamento não volátil, opera até 32 MHz e inclui periféricos como USART, MSSP, PWM e ADC de 10 bits.

O dispositivo conta com comparadores analógicos, interrupções prioritárias, modos de baixo consumo de energia e funções de segurança. Seus pinos de I/O proporcionam flexibilidade, e está disponível em diversos encapsulamentos. A consulta à loja de dados é essencial para detalhes elétricos e exemplos de aplicação.

Circuito eletrônico com fios

Descrição gerada automaticamente

Figura 2 - Microcontrolador PIC 16F1619

Fonte: [Microchip Technology](https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.microchip.com%2Fen-us%2Fproduct%2Fpic16f1619&psig=AOvVaw20AJwEg7cAsRaDQyiCYzL8&ust=1700626452838000&source=images&cd=vfe&opi=89978449&ved=0CAUQjB1qFwoTCPjGy62d1IIDFQAAAAAdAAAAABAE)

**Motor DC de 3 a 6 volts com caixa de redução e eixo duplo**

O motor DC de 3 a 6 volts com caixa de redução e eixo duplo é um dispositivo elétrico projetado para converter energia elétrica em movimento mecânico, proporcionando torque e rotação controlada. Este tipo específico de motor é projetado para operar em uma faixa de tensão entre 3 e 6 volts, tornando-o adequado para uma variedade de aplicações que requerem um motor de baixa voltagem.

A presença de uma caixa de redução é uma característica significativa deste motor. Essa caixa desempenha o papel de reduzir a velocidade de rotação do motor, aumentando simultaneamente o torque disponível. Isso é especialmente útil em situações em que é necessário um maior poder de condução e controle de movimento preciso.

O eixo duplo refere-se à presença de dois eixos de saída no motor. Isso proporciona mais versatilidade e flexibilidade na utilização do motor em diferentes aplicações. Os eixos duplos podem ser empregados para acionar dispositivos mecânicos em ambos os lados do motor, possibilitando uma maior variedade de configurações e aplicações práticas.

Uma imagem contendo lego

Descrição gerada automaticamente

Figura 3 - Motor DC de 3 a 6 volts com caixa de redução e eixo duplo

*Fonte: Robocore*

**Carrinho conversível**

# Este estudo empregou um modelo [Lamborghini Gallardo LP 560-4](https://brickset.com/sets/8169-1/Lamborghini-Gallardo-LP-560-4) de LEGO. A integração dos eixos das rodas com o eixo do motor foi realizada, permitindo a movimentação coordenada desses componentes no projeto.

Carro de brinquedo

Descrição gerada automaticamente com confiança média

Figura 4 - Lamborghini Conversível de LEGO

*Fonte: TheCoolist*

**Driver ponte H L298N**

O Driver Ponte H L298N é um componente eletrônico amplamente utilizado para controle de motores em projetos de robótica e automação. Projetado para operar como uma ponte H, o L298N permite controlar a direção e a velocidade de motores DC e motores de passo. Com sua capacidade de fornecer correntes mais altas e suportar tensões variadas, é uma escolha popular para aplicações que exigem precisão no controle de motores.

Este dispositivo é composto por duas pontes H independentes, cada uma capaz de controlar um motor, proporcionando reversibilidade e controle bidirecional. Além disso, o L298N inclui proteções internas contra sobretensão e sobrecorrente, aumentando a confiabilidade e a durabilidade em ambientes de trabalho diversos. Sua interface simples e compatibilidade com microcontroladores tornam o L298N uma escolha versátil para projetos que demandam eficiência no controle de motores.

Circuito eletrônico com fios

Descrição gerada automaticamente com confiança média

Figura 5 - Driver Ponte H L298N

*Fonte: Eletrodex*

**Adaptador CP2102 USB-SERIAL TTL**

O componente CP2102 USB-SERIAL TTL é um dispositivo crucial na interconexão entre dispositivos eletrônicos e computadores por meio da interface USB. Ele desempenha um papel fundamental na conversão de dados entre o formato serial TTL (Transistor-Transistor Logic) e o protocolo USB, facilitando a comunicação eficiente entre microcontroladores e computadores.

O CP2102 USB-SERIAL TTL foi empregado para possibilitar a transmissão de informações do PIC para a porta serial do Arduino. Essa implementação permitiu a análise em tempo real das informações relativas ao carrinho, destacando a eficácia do componente na facilitação da comunicação entre os dispositivos eletrônicos envolvidos no projeto.

Circuito eletrônico em superfície de madeira

Descrição gerada automaticamente com confiança baixa

Figura 6 - Adaptador CP2102 USB-SERIAL TTL

**Métodos**

**Desenvolvimento**

Após a escolha do projeto, foi decidido utilizar um carrinho LEGO composto por chassi e rodas, porém sem motor. Contudo, um desafio surgiu devido à incompatibilidade do eixo do motor com o sistema do carrinho. Para contornar essa limitação, foram fixadas quatro placas de MDF de 3mm cada. Nestas placas, foram feitos furos de 6mm no formato do eixo do motor e outro furo na outra face de 6mm no formato do eixo da roda. Essa adaptação eficiente solucionou o problema de conexão, permitindo a integração adequada do motor ao carrinho e assegurando a viabilidade do projeto.

Durante a primeira montagem do carrinho foi utilizado um Arduino Uno ao invés do PIC16f1619 pela facilidade que o Arduino proporciona no momento de programar o código. Assim foi criado um código simples que utilizava da logica *if* para fazer o carro se mexer de acordo com o a distância estabelecida, que foi decidida em 20 centímetros. A partir dos dados gerados com o funcionamento do carrinho utilizando o Arduino foi possível criar uma função de amortecimento utilizando o LabView para fazer o tratamento dos dados e a fórmula de amortecimento, que em seguida foi enviada ao MatLab para fazer a fórmula capaz de integrar no código. O primeiro controle utilizado no projeto foi um PID, porém, por conta da zona morta do motor escolhido, resultava em overshoot durante os testes e assim foi necessário mudar o controle para um PD, que após os testes demonstrou funcionar bem e com a precisão necessária para o projeto.

Quando o resultado utilizando o Arduino se mostrou satisfatório para o projeto, faltava apenas adaptar código já em funcionamento no Arduino para o PIC, que demonstrou ser mais desafiador do que esperado.

Durante o desenvolvimento deste projeto, um dos desafios enfrentados foi a necessidade de empregar, no microcontrolador PIC, uma ferramenta semelhante ao "millis" disponível no Arduino. O objetivo era coletar dados de distância provenientes de um sensor ultrassônico. No contexto do Arduino, a função "millis" é intrínseca, incrementando uma variável a cada milissegundo. Esta funcionalidade permitiu, por meio da contagem temporal resultante, determinar o intervalo entre a emissão e o retorno do sinal ultrassônico. Ao multiplicar este intervalo por uma constante predefinida, foi possível obter a medida precisa da distância entre o sensor e o objeto em questão.

Para adaptar essa abordagem para o microcontrolador PIC, foi necessário a utilização da configuração de interrupções externas, que possibilita a interrupção de tarefas regulares para executar uma tarefa de prioridade mais alta quando ocorre um evento externo. Essa configuração junto com o Cristal de microcontrolador, possibilitam uma versão do millis no PIC.

Depois de solucionar o problema do tempo para o sensor ultrassónico, adaptar o resto do código foi mais simples. Para configuração do PWM foi necessário ajustar um timer para enviar os pulsos de tensão na velocidade ideal o funcionamento do mesmo, quando finalizado o PWM estava funcionando como esperado.

Para a finalização do código no PIC exigiu a configuração do EUSART para que os dados pudessem ser vistos no monitor serial do Arduino IDE. Assim que foi configurado no MCC do MPlab, apenas foi necessário adicionar a ordem *printf* no código que dados requisitados eram reconhecidos e enviados para o monitor serial.

**Modelagem do Controle**

Primeiramente no desenvolvimento do sistema de controle foram necessários registrar os dados do projeto funcionando apenas com a lógica do *if* que consistia em fazer o carrinho funcionar e recolher os dados do momento em que ele começa a desacelerar até o momento da parada final. Para fazer a colheita dos dados utilizamos a porta serial do Arduino que enviava os dados para um modelo do LabView modificado.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Figura 7 - Interface do modelo modificado do LabView

Após a colheita dos dados eles foram enviados para uma planilha do Excel onde foram tratados antes de serem enviados de volta ao LabView. No Excel os dados foram invertidos para facilitar a utilização deles. Como é possível observar na imagem abaixo:

Gráfico

Descrição gerada automaticamente

Figura 8 - Planilha do Excel com os dados puros e invertidos

Depois de inverter os dados eles foram enviados de volta para o LabView onde foi possível fazer um gráfico de velocidade a partir do primeiro, de distância. Isso foi feito para conseguir a função de transferência da velocidade que melhor ajustaria ao projeto.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Figura 9 - Identificação da função de transferência

Diagrama, Esquemático

Descrição gerada automaticamente

Figura 10 - Back-end da interface da imagem acima

Primeiramente houve a tentativa de utilizar um PID para controlar a velocidade do projeto, porém, depois de testes com o sistema adicionado no carrinho, era possível perceber problemas com overshoot, que se originavam do motor escolhido para o projeto, que possuía uma zona morta que influenciava no compensador resultando na eficiência do controle.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Figura 11 - Controlador PID

Assim foi necessário alterar o compensador para um PD simples, que permitiu um funcionamento perfeito do carrinho. Interface gráfica do usuário, Gráfico, Histograma

Descrição gerada automaticamente Figura 13 - Gráficos da função de transferência

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Figura 12 - Controlador PD

Com a função de transferência pronta, foi necessário descobrir o controle em Z para ser possível que a função de transferência possa ser colocada no código do Arduino e posteriormente, no PIC16f1619.

Como o controlador PD resultou em apenas uma constante, a sua aplicação se tornou simples.

Diagrama

Descrição gerada automaticamente

A ilustração acima representa o sistema de malha fechada utilizado. O "processo" corresponde aos dados coletados e processados pelo LabView gerando a função de transferência que representa o funcionamento do motor do carrinho. O "sensor de medida" é representado pelos valores obtidos a partir do sensor ultrassônico, expressos em centímetros. A "Referência" é um valor predefinido no código, também em centímetros. O controle é realizado pelo MATLAB, que gera um valor de controle com uma constante de 35,51.

O funcionamento do sistema ocorre através da retroalimentação do processo (motor) com um valor em PWM. Esse valor é a referência subtraída do sensor de medida (sensor ultrassônico), e o resultado é multiplicado pela constante do controlador. O valor resultante é então enviado para o motor em termos de PWM no código desenvolvido.

**Configurações do PIC16f1619 que foi utilizada no projeto:**

**Configuração do PIC**

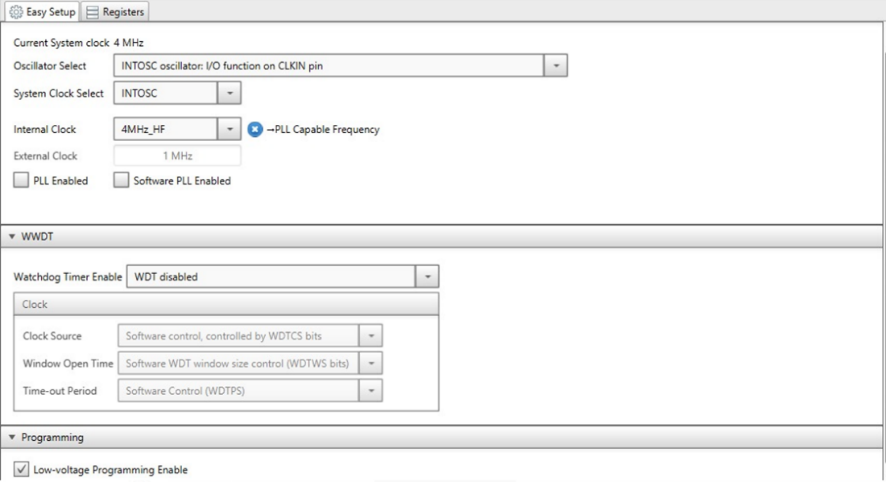


Figura 14 - Configuração do PIC no MCC

Configuração das saídas e entradas no PIC**:**

No âmbito do Projeto de Iniciação Científica (PIC), é necessário que cada uma das saídas e entradas seja devidamente configurada para assegurar sua utilização adequada. Nesse contexto, o sensor ultrassônico foi configurado nas portas RA4 e RA5. A porta RA4, denominada ECHO, foi configurada como saída, sendo responsável pelo envio do sinal de onda sonora, enquanto a porta RA5, denominada TRIGGER, foi configurada como entrada, responsável por receber o referido sinal.

A ponte H, componente crucial para o controle do motor, foi configurada nos pinos RC4, RC5 e no RA0. As duas primeiras portas, RC4 e RC5, desempenham o papel de emitir sinais para determinar a direção em que o motor opera, enquanto a última, RA0, é responsável por definir o valor do sinal PWM aplicado ao motor. Essas configurações são essenciais para o correto funcionamento do sistema, garantindo uma integração eficiente entre os componentes envolvidos no projeto.

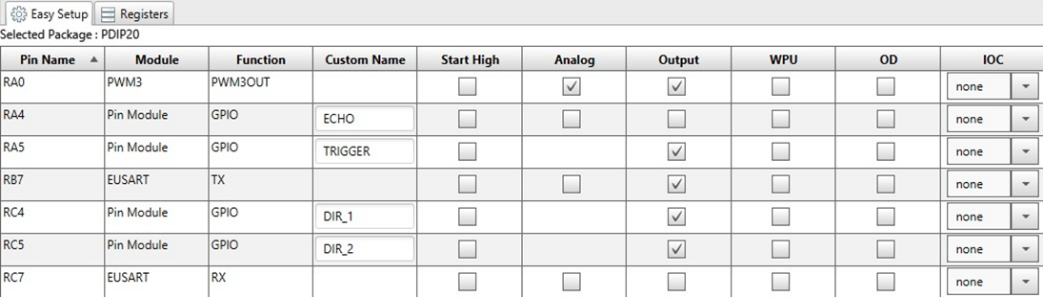


Figura 15 - Configurações dos pinos no MCC

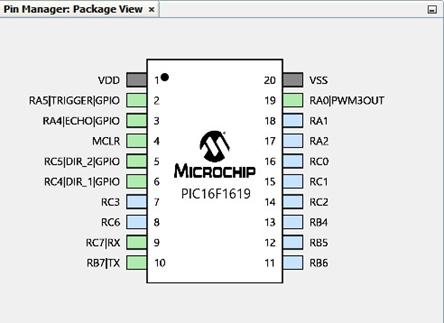
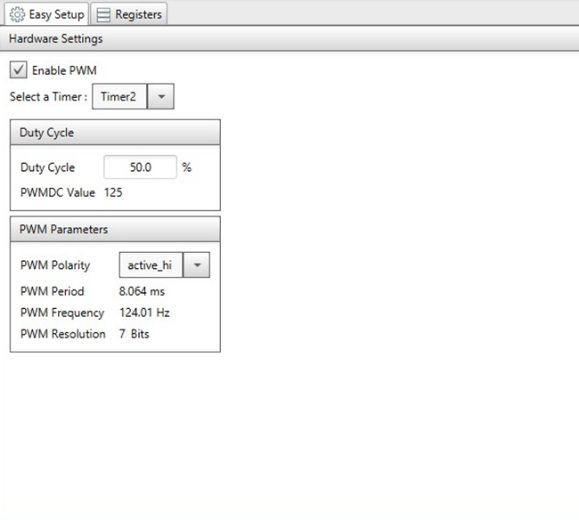


Figura 16 - Pin Manager do PIC

Configuração do timer e o PWM para a aplicação:

Interface gráfica do usuário

Descrição gerada automaticamente

Figura 17 - Configuração do timer no MCC Figura 18 - Configuração do PWM no MCC

Configuração do EUSART para que fosse possível realizar a conexão Serial entre o PIC e o monitor serial do Arduino IDE no computador:

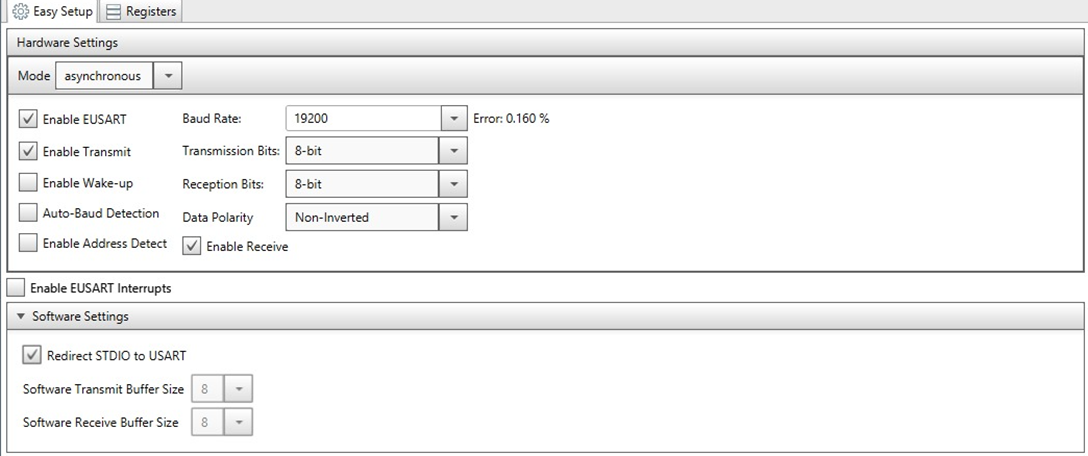


Figura 19 - Configuração do EUSART no MCC

**Utilização do sensor HC-SR04 no PIC**

Ao contrário do Arduino, o PIC não dispõe de bibliotecas específicas para a obtenção direta da distância a partir de um sensor ultrassônico. Nesse contexto, foi essencial desenvolver uma lógica de programação que calculasse o tempo decorrido entre o envio e a recepção do sinal, proporcionando a obtenção do valor desejado. O código implementado para essa finalidade é apresentado abaixo:

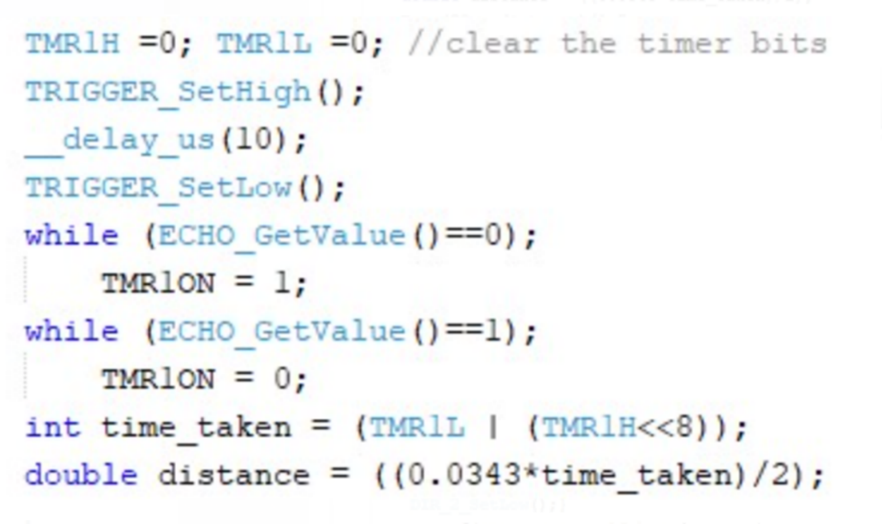


Figura 20 - Programação do sensor ultrassônico

No âmbito dos microcontroladores, os cristais desempenham um papel crucial ao emitirem um sinal de alta frequência, sincronizando as funções temporais com o clock do cristal. Entretanto, surge um desafio quando as funções temporais registram valores superiores a 256, resultando no que é conhecido como "timer overflow" e provocando o reset da contagem, desencadeando uma interrupção externa.

Contudo, considerando que os cristais oscilam a uma frequência de 4 MHz, o que poderia resultar em milhares de timer overflows por segundo, os microcontroladores adotam uma abordagem inteligente. Eles "pulam" oscilações para acumular a contagem, gerando interrupções externas em intervalos que podem ser personalizados. É nesse contexto que surge a oportunidade de ajustar os valores para utilizar uma função análoga ao "Millis" do Arduino, proporcionando uma maior flexibilidade e controle sobre as interrupções externas, tornando-as mais adequadas às necessidades específicas do sistema em questão.

**Tratamento da saída e logica para ponte H**

O controle do carrinho foi implementado por meio de um controlador PD, com o objetivo de amortecer a resposta do sistema e assim suavizar o movimento do veículo, que também é acompanhado por lógicas condicionais (IF) específicas. Essas lógicas foram incorporadas com o intuito de minimizar as áreas de Overshoot e resolver problemas relacionados à saída, promovendo assim uma estabilização eficaz. O código criado e utilizado para esta aplicação pode ser visto na figura a seguir.

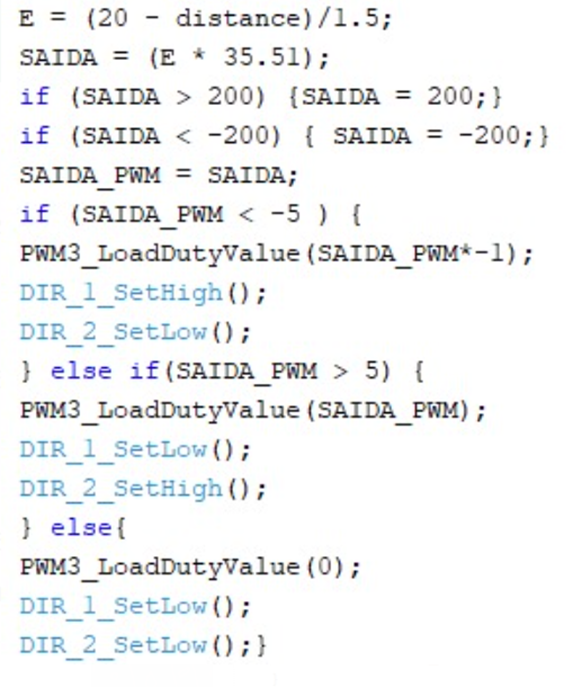


Figura 21 - Programação do controle e saída do motor

Visando um controle melhor do carrinho, a saída, que o valor também é utilizado como PWM, foi limitada a valores entre 5 e 200 para evitar valores superiores aos necessários para o funcionamento da ponte H e valores dentro da zona morta do motor. Além disso, foi implementada uma lógica condicional adicional (IF) para interpretar valores negativos, permitindo a movimentação para frente e para trás conforme a lógica da ponte H.

**Utilização do EUSART e Adaptador USB-SERIAL TTL**

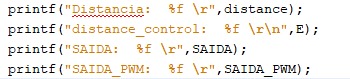


Figura 22 - Programação do que aparece no monitor serial

Após a adição do *printf* no código o monitor serial do Arduino IDE, conectado no Adaptador CP2102 USB-SERIAL TTL via USB, recebia os dados enviados pelo PIC, que no caso mostra a distância (distance) em q o carrinho estava do objeto, o erro (E), que é a diferencia entre a distância e os 20 centímetros, a resposta do controle (SAIDA) e o valor em PWM (SAIDA\_PWM) que o código estava enviado ao motor.

# **Resultados e Discussão**

Após a conclusão do projeto os resultados apresentados foram muito satisfatórios, mantendo o carrinho sempre a uma distância de 20 centímetros do objeto a sua frente e com o controle de amortecimento funcionado perfeitamente para fazer os ajustes de finais de forma suave e precisa. Infelizmente não é possível demostrar os resultados por esse relatório.

Para uma possível continuação do projeto seria interessante diminuir o tempo de reação do carrinho para que ele possa se posicionar mais rapidamente na distância alvo.



Figura 23 - Projeto finalizado

### **Conclusões**

Na conclusão deste projeto, destacam-se os resultados alcançados na implementação do Lambo Control, com foco nas áreas de Instrumentação, Microcontroladores e Sistemas de Controle. A interação harmoniosa desses componentes desempenhou um papel essencial na efetivação do projeto, evidenciando a aplicação prática dos conceitos teóricos em um contexto operacional. Ao explorar as funcionalidades do sensor ultrassônico, microcontrolador PIC 16F1619 e controlador PD, este projeto proporcionou uma visão objetiva sobre a integração e aplicação dessas tecnologias em sistemas autônomos controlados eletronicamente.

Na área de Instrumentação, o sensor ultrassônico HC SR04 foi eficaz na medição precisa da distância entre o carro autônomo e o alvo, contribuindo para um controle preciso do sistema.

No contexto dos Microcontroladores, o microcontrolador PIC 16F1619 foi escolhido para garantir a operação eficiente do Lambo Control, fornecendo capacidade de processamento e comunicação adequadas. O adaptador CP2102 USB-Serial TTL facilitou a comunicação entre o microcontrolador e o sistema de controle.

Quanto aos Sistemas de Controle, a implementação de um controlador PD foi adotada, proporcionando uma resposta controlada ao sistema e evitando problemas de overshoot. A utilização de uma lógica if simples para a realimentação do motor mostrou-se eficaz na prevenção de colisões.

Em resumo, a aplicação prática desses componentes em suas respectivas áreas contribuiu para o sucesso do Lambo Control, fornecendo dados precisos, operação eficiente e controle adequado do sistema. Esses resultados são observáveis nos aspectos práticos da implementação do projeto.

### **Referências Bibliográficas**

Interrupção externa. Docs.franzininho.com.br. Disponível em: <https://docs.franzininho.com.br/en/docs/franzininho-diy/exemplos-avr-libc/int-ext/#:~:text=As%20interrup%C3%A7%C3%B5es%20externas%20permitem%20que,quando%20ocorre%20um%20evento%20externo>. Acesso em: 20/11

How to use pic microcontroller timers as a interrupt. [microcontrollerslab.com](https://microcontrollerslab.com). Disponível em: <https://microcontrollerslab.com/how-use-pic-microcontroller-timers-interrupt/#google_vignette>. Acesso em: 20/11.