	Universidade Federal Rural de Pernambuco Unidade Acadêmica de Belo Jardim Engenharia da Computação	
	Relatório Projeto Final – Sistema detector de metais com atuador para descarte de materiais e não metálicos.	
Disciplina:	UAB00093 - Engenharia de Sistemas Embarcados	
Professor:	Henrique Patriota	
Alunos:	Marcelo Augusto de Barros Araújo / Pedro Henrique Lima Pereira.	
Simulação/ Bancada:	Tinkercad e feito na prática	Data: 11/03/2025

1. Introdução

O objetivo desse projeto é criar um Sistema capaz de detectar metais com uso de atuador para descarte de materiais metálicos e não metálicos, conhecido também por segregador de materiais que na automação industrial tem desempenho fundamental na otimização de processos de manufatura e reciclagem. Nesse contexto, a detecção e segregação automático de materiais tem um papel crucial na eficiência dos sistemas de triagem. Este projeto tem como objetivo desenvolver um sistema embarcado capaz de identificar materiais metálicos e não metálicos, realizando seu descarte adequado por meio de um atuador servo-motor. O sistema, conhecido como segregador de materiais, faz uso de sensores, atuadores e técnicas de programação de baixo nível para garantir um funcionamento rápido e confiável.

Esse tipo de aplicação de automação industrial é muito utilizado na indústria, para separação de materiais como em esteiras de grandes fábricas e até mesmo em cooperativas de reciclagem de material, esse projeto tem a visão de criar um protótipo de baixo custo que possa ser replicado futuramente em ambientes reais de produção, com algumas melhorias como por exemplo display LCD para melhor visualização, banco de dados para acompanhar o número de materiais metálicos identificados, watchdog timer para resets automáticos em caso de falha, modo sleep para economia de energia e entre outras melhorias que pode ser implementado nesse protótipo futuramente.

A implementação utiliza microcontroladores programados em linguagem C e explora o uso de registradores em substituição ao comando pinMode da IDE do arduino, garantindo maior eficiência na manipulação dos periféricos do microcontrolador. O uso direto dos registradores permite uma melhor otimização do código, reduzindo o tempo de execução e o consumo de recursos do sistema. Além disso, o projeto faz uso de interrupções externas e internas para garantir uma resposta rápida à detecção de metais e ao controle do tempo de movimentação do servo-motor, uma vez que os

registradores operam com um ciclo de máquina mais eficiente do que as abstrações fornecidas pela IDE do Arduino.

Além disso, o projeto faz uso de interrupções internas e externas para garantir uma resposta rápida à detecção de metais e ao controle do tempo de movimentação do servo-motor. O uso de técnicas como a configuração de prescalers e o ajuste do registrador TCNT no Timer permite um controle preciso dos eventos com menor latência e um consumo reduzido de processamento. Essa abordagem melhora o desempenho do sistema, pois evita a necessidade de polling constante, liberando a CPU para outras tarefas e reduzindo o consumo de energia.

O sistema é composto por um sensor ultrassônico para detecção de objetos, um sensor indutivo de metal para identificação do material metálico e um servo-motor responsável pelo descarte adequado dos itens. A lógica de funcionamento baseia-se na leitura dos sensores, no processamento dos sinais e na tomada de decisão automatizada para a movimentação do servo-motor. Enfim uma das técnicas que possibilita esse ganho com aplicação de registradores é a de BitWise, pois o uso de operações como $|$ (OR bit a bit) para configurar bits específicos, $\&$ (AND bit a bit) para limpar bits e \wedge (XOR bit a bit) possibilitando essa alternância e gerando um controle mais eficiente de entrada/saída dos periféricos, reduzindo o overhead e tornando o código mais compacto e otimizado, nesse projeto foi utilizado o setBit que um 'ou' lógico que altera o nível entre 0 e 1.

2. Objetivos

2.1. Objetivo Geral

Desenvolver um sistema embarcado de detecção e segregação de materiais metálicos e não metálicos, utilizando o Arduino Uno com o microcontrolador ATmega328P embarcado como unidade de processamento principal. O projeto faz uso de registradores, interrupções internas para controle de eventos de espera entre uma ação e outra, como aplicado nessa prática, onde o servo só realiza o movimento de metal, após dois segundos, já para não metal após 4 segundos e externas para controle do sensor de metal e operações bitwise, otimizando o desempenho e reduzindo o consumo de memória. A integração de sensores e o atuador combinado a interrupção externa INT0 no ATmega328P garante a detecção de metais sem a necessidade de verificação constante dentro do loop(). Isso garante que, assim que o sensor de metal identifica um material metálico, o microcontrolador responde imediatamente, evitando atrasos causados por verificações manuais. Essa técnica melhora a eficiência do sistema, reduz o consumo de energia e libera o processador para outras tarefas.

Além disso, a interrupção interna por Timer2 substitui o uso de delay(), permitindo que o tempo de espera entre a detecção do objeto e a movimentação do servo seja controlado sem

bloquear a execução do código. O prescaler e a manipulação do TCNT2 garantem precisão no tempo de resposta, otimizando o funcionamento do sistema.

Figura 1 – Tabela de Prescales

CS02	CS01	CS00	Description
0	0	0	No Clock Source
0	0	1	System Clock
0	1	0	Prescaler = 8
0	1	1	Prescaler = 64
1	0	0	Prescaler = 256
1	0	1	Prescaler = 1024
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

2.2. Objetivos Específicos

- Utilizar o Arduino Uno como plataforma de desenvolvimento, explorando os recursos do ATmega328P, incluindo manipulação direta de registradores e otimização do tempo de execução.
- Empregar uma protoboard para prototipação do circuito, facilitando testes e ajustes nos componentes para implementação final.
- Implementar um sensor de metal para detecção de materiais metálicos, acionando interrupções externas para resposta imediata ao evento.
- Utilizar um sensor ultrassônico para identificação da presença de objetos na área de verificação, disparando o processo de segregação.
- Aplicar manipulação direta de registradores para configurar os pinos de entrada e saída do ATmega328P, substituindo funções de alto nível como pinMode() e digitalWrite(), reduzindo o tempo de execução.
- Explorar operações bitwise para otimização do controle dos registradores, garantindo menor latência e baixo consumo de memória.

- Configurar interrupções internas via Timer com prescaler e TCNT para criar eventos periódicos de controle do sistema, sem a necessidade de verificações constantes no loop().
- Integrar um servo motor para segregação automática, acionando a posição correta de descarte de acordo com a classificação do material detectado.
- Maximizar a eficiência energética e de processamento, evitando operações bloqueantes e priorizando eventos assíncronos via interrupções.
- Foi usado o Timer2 do ATmega328P é um timer de 8 bits, ou seja, conta de 0 até 255 antes de "transbordar" (overflow) e disparar a interrupção, ao iniciar TCNT2 com 5, garantimos que o timer gere uma interrupção a cada 1ms exato, evitando atrasos cumulativos, onde a frequência do clock do Arduino Uno é 16 MHz, e o prescaler de 64 reduz essa frequência é: 250.000 Hz.

3. Metodologia

Materiais e Equipamentos:

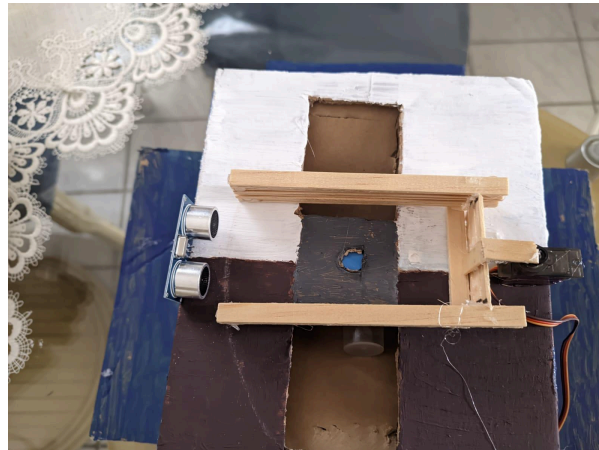
- Caixa de papelão com 20 cm de comprimento, 16 cm de largura e 10,9 cm de altura.

Figura 2 – Caixa de Papelão



- Dois buracos na caixa, com dimensões de 6,5 cm de comprimento e 4,5 cm de largura. Tinta, cola quente, tesoura e estilete para montagem da estrutura.
- Paletas de picolé de 12 cm de comprimento e largura 0,5 cm, onde foi necessário 24 paletas no total para montar a base seletora do servo (dispostas nas laterais esquerda e direita da caixa) e pequenas paletas para suportar o servo no centro da caixa, além de um pequeno furo no centro da caixa para suporte do sensor de metal.

Figura 3 – Caixa com base e sensores dispostos



- ☐ 1 Arduino Uno (modelo ATmega328P).
- ☐ 2 LEDs (um verde e um vermelho) com dois resistores de 220 Ω .
- ☐ 1 Sensor Indutivo NPN de Proximidade LJ12A3-4-Z/BX (Normalmente Aberto), com um resistor de 10 k Ω .
- ☐ 1 Sensor Ultrassônico HC-SR04.
- ☐ 1 Bateria de 9V para alimentação do sensor indutivo que funciona com alimentação entre 6 e 36 dC, mas para leitura correta do sinal é recomendado no mínimo 9V.
- ☐ 1 Conector P4 fêmea para conexão do cabo adaptador da bateria.
- ☐ 1 Cabo adaptador de alimentação para a bateria.
- ☐ 1 Protoboard para prototipação do circuito.
- ☐ 1 Computador com entrada USB para alimentar o arduino e programar o microcontrolador.
- ☐ Vários jumpers Macho-Fêmea e jumpers Macho/Macho para conexão na protoboard/arduino, além de jumpers Fêmea-Fêmea para extensão da conexão.

Procedimento de Montagem:

Estrutura física: A caixa de papelão foi cortada e adaptada para criar os buracos necessários para a inserção de componentes, como os sensores e a base seletora. A cola quente foi usada para fixar as paletas de picolé nas laterais e garantir estabilidade à estrutura.

Montagem da base seletora do servo: Usando as 9 paletas de picolé, foi construída a base seletora do servo na parte lateral da caixa. As 6 paletas menores foram usadas para criar um suporte firme para o servo, permitindo que ele fosse fixado com precisão.

Circuito eletrônico: O Arduino Uno foi programado para controlar os LEDs, o sensor indutivo e o sensor ultrassônico. Os LEDs foram conectados aos pinos digitais do Arduino e os resistores de 220 ohms foram utilizados para limitar a corrente e proteger os LEDs, além do resistor de 10 kohms para o sensor indutivo e utilização de uma protoboard para prototipar o circuito.

Instalação dos sensores: O sensor indutivo foi instalado na posição estratégica para detectar objetos metálicos próximos, já que ele detecta em um raio muito próximo de 4mm. O sensor ultrassônico foi posicionado para medir distâncias, coletando informações sobre a proximidade de objetos em uma distância de 10,5 cm entre o sensor ultrassônico e a base do servo em um ângulo central, deixando a área de percepção livre para identificar o objeto.

Fonte de alimentação: A bateria de 9V foi conectada ao Arduino através do conector P4 fêmea, garantindo que o sistema fosse alimentado adequadamente durante o funcionamento.

Ajustes e Calibração:

O código de controle do Arduino foi ajustado para garantir a correta interação entre os sensores e os LEDs.

Foi necessário calibrar o sensor ultrassônico para obter medições precisas de distância, ajustando a leitura dos valores de tempo de resposta.

A base seletora foi projetada para permitir movimento suave, garantindo que o servo respondesse adequadamente ao controle de entrada.

Método de Medição:

O sensor ultrassônico foi utilizado para medir a distância de objetos, e o valor obtido foi exibido no monitor serial do Arduino mostrando a distância entre o objeto e o sensor, além de informar o movimento junto ao tempo de espera entre cada ciclo de movimento do servo motor.

O sensor indutivo foi utilizado para detectar a proximidade de objetos metálicos, acionado o LED verde quando detectado e o LED vermelho quando não detectado com auxílio do ultrassônico.

A verificação do funcionamento do sistema foi realizada verificando se os LEDs acendiam corretamente e se a movimentação do servo ocorria de forma adequada conforme os estímulos recebidos pelos sensores.

Segue abaixo o esquema de montagem feito no simulador Tinkercad e o esquema de montagem feito na prática

4. Resultados e Discussões

O projeto foi desenvolvido para automatizar a detecção de objetos não metálicos e a identificação de materiais metálicos, utilizando um servo motor, um sensor ultrassônico e um sensor de metal, controlados por um microcontrolador ATmega328P (base do Arduino). Abaixo, detalho os resultados encontrados durante os testes do sistema e como cada sensor foi utilizado no projeto:

1. Sensor Ultrassônico (HC-SR04)

O sensor ultrassônico tem como principal função medir a distância entre o sensor e um objeto. Ele funciona enviando ondas sonoras de alta frequência e aguardando o tempo que elas demoram para retornar após refletirem em um objeto. A partir deste tempo, é possível calcular a distância com base na velocidade do som no ar, onde o sensor emite uma onda de ultrassom através do pino TRIG, quando a onda atinge um objeto, ela retorna ao sensor e é captada pelo pino ECHO.

Durante o teste, o sensor foi capaz de detectar objetos em um intervalo de 0 a 8 cm de distância, mas na prática mesmo seria a partir de 2cm segundo a especificação do sensor conforme o resultado esperado. Quando um objeto foi detectado dentro dessa faixa de distância, o sistema registrou e começou a esperar pela detecção de metal. O sensor apresentou uma precisão razoável tendo uma variação acima de 1cm e o tempo de resposta foi adequado para o controle do servo motor, vale ressaltar que também foi utilizada comunicação serial através do protocolo UART para monitorar o movimento do servo com mensagens de forma assíncrona.

2. Sensor de Metal (Detecção de Metal por Interrupção Externa)

O sensor de metal foi configurado para detectar a presença de um objeto metálico próximo ao dispositivo. A detecção é feita através de um pino de interrupção do microcontrolador, que é ativado quando o sensor detecta um objeto metálico. Essa interrupção é utilizada para controlar o movimento do servo motor, alterando sua posição dependendo da detecção.

Funcionamento:

O sensor de metal é um tipo de interruptor digital que envia um sinal de nível baixo (LOW) ou nível alto (HIGH) quando detecta a presença de metal.

Quando o metal é detectado, uma interrupção externa (configurada no pino INT0) é acionada e marca o estado da detecção do metal, fazendo com que o servo motor seja movimentado para a posição desejada.

Resultados obtidos: Durante os testes, o sensor de metal funcionou corretamente, acionando a interrupção externa quando um objeto metálico foi detectado. O sistema foi projetado para esperar a

detecção de metal dentro de um tempo máximo de 4 segundos. Caso o metal fosse detectado antes disso, o servo motor se moveria para a direita (para 45°); caso contrário, o servo motor se moveria para a esquerda (para 135°).

3. Controle do Servo Motor(Detecção de não Metal por Interrupções internas)

O servo motor foi utilizado para mover uma parte do sistema com base na detecção dos sensores. O controle do servo foi implementado com a biblioteca Servo, que permite mover o motor para uma posição específica.

Funcionamento:

O servo motor foi configurado para mover-se para 3 posições diferentes: 90° (posição central-inicial), 45° (movimento para a direita-metal), e 135° (movimento para a esquerda-não metal).

O servo foi acionado dependendo do resultado da detecção do objeto e do metal. Se o objeto foi detectado e o metal foi encontrado dentro do tempo esperado, o servo se move para 45°; caso contrário, ele se move para 135°.

Resultados obtidos:

Quando o metal foi detectado no tempo esperado (2 segundos), o servo motor foi corretamente movido para 45°, indicando que o metal foi encontrado.

Quando o metal não foi detectado dentro do tempo máximo de 4 segundos, o servo motor foi movido para 135°.

Após 700 ms, o servo voltou à posição inicial (90°), e o sistema aguardou por uma nova detecção de objeto.

Resultados da Simulação e Comparação com Teoria

Para este projeto, a simulação de funcionamento do sistema foi realizada utilizando um ambiente de prototipagem como o Tinkercad para montagem do hardware mais não foi possível testar todos os componentes em simulação, pois dependia de alguns objetos da vida real para teste do projeto, onde a interação entre o sensor ultrassônico, o sensor de metal e o servo motor foi verificada na prática testando cada um de forma individual o seu funcionamento. Por que na simulação foi possível comprovar uma certa limitação de sensores e testes que eram necessários para implementação do projeto.

Ao comparar os resultados da simulação de montagem de hardware e os resultados práticos, observou-se uma boa correspondência, com o tempo de resposta do sensor ultrassônico e o sensor de metal sendo consistente os estouros do timer2 definidos. A principal diferença foi um pequeno atraso na resposta do servo motor, principalmente na hora de montar a base, pois na primeira

tentativa de montar a base o mesmo não estava movendo por causa do peso, então foi necessário testar primeiro com papelão antes da base final ser ajustada para paletas de picolé.

Segue abaixo o esquema de montagem feito na simulação, o código utilizado e a montagem prática do projeto:

Figura 4 – Esquema de montagem na Simulação

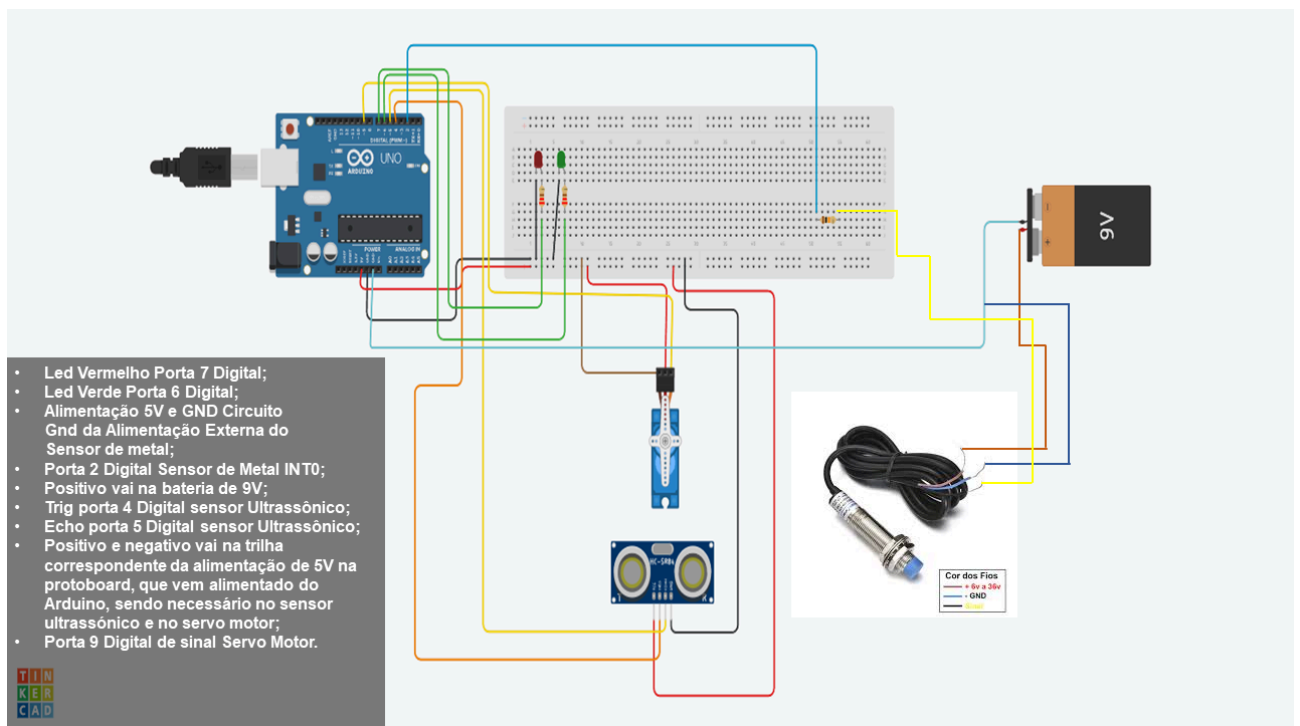


Figura 5 – Esquema de montagem na Prática

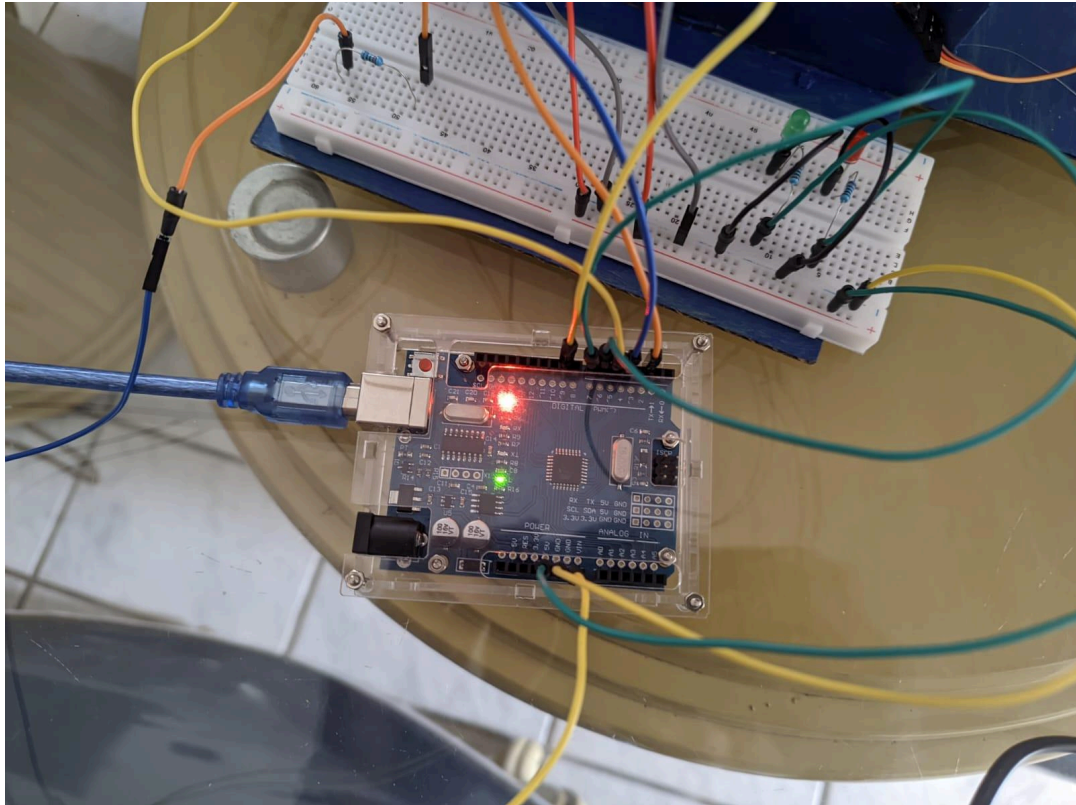


Figura 6 – Esquema de passagem dos fios do Servo e Ultrassônico

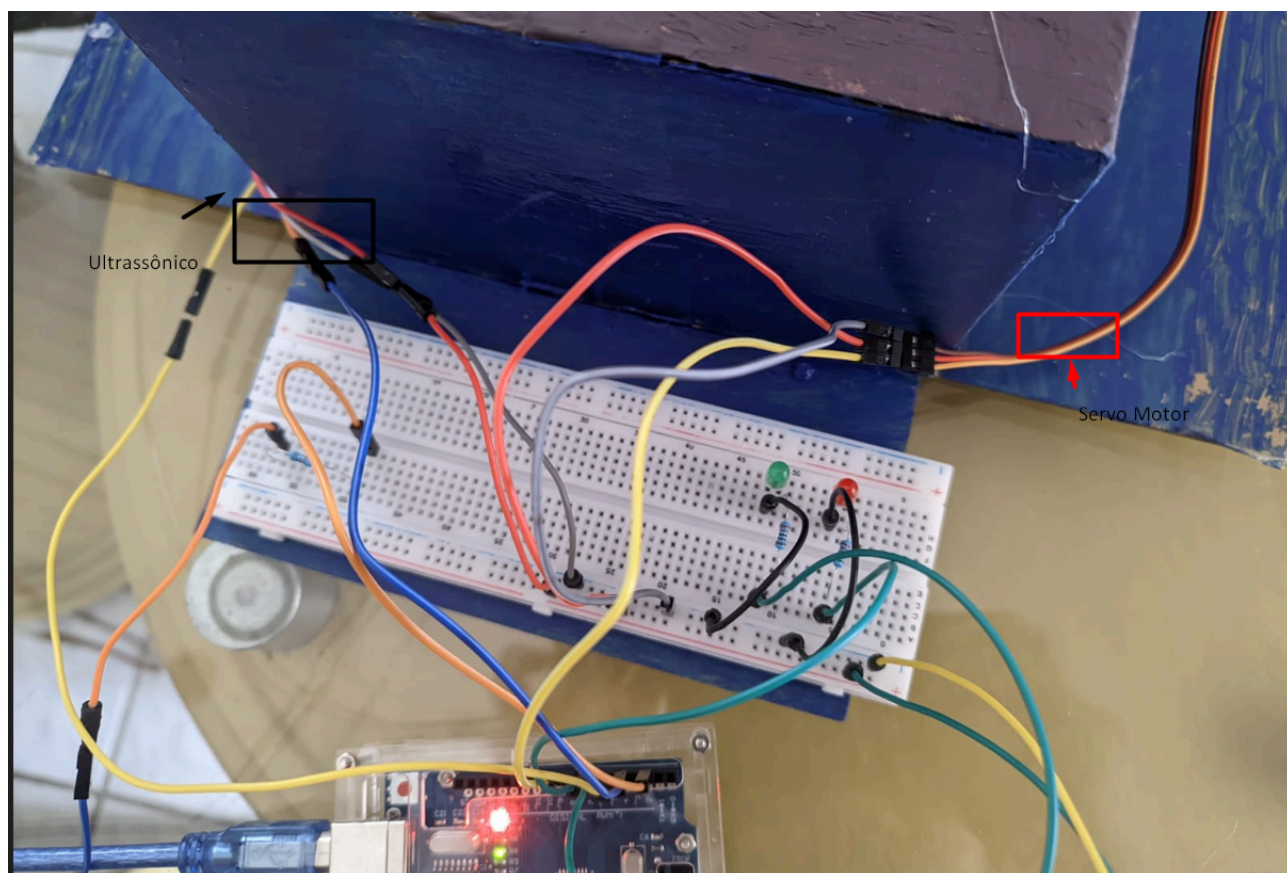


Figura 7 – Esquema de posicionamento Sensor Ultrassônico

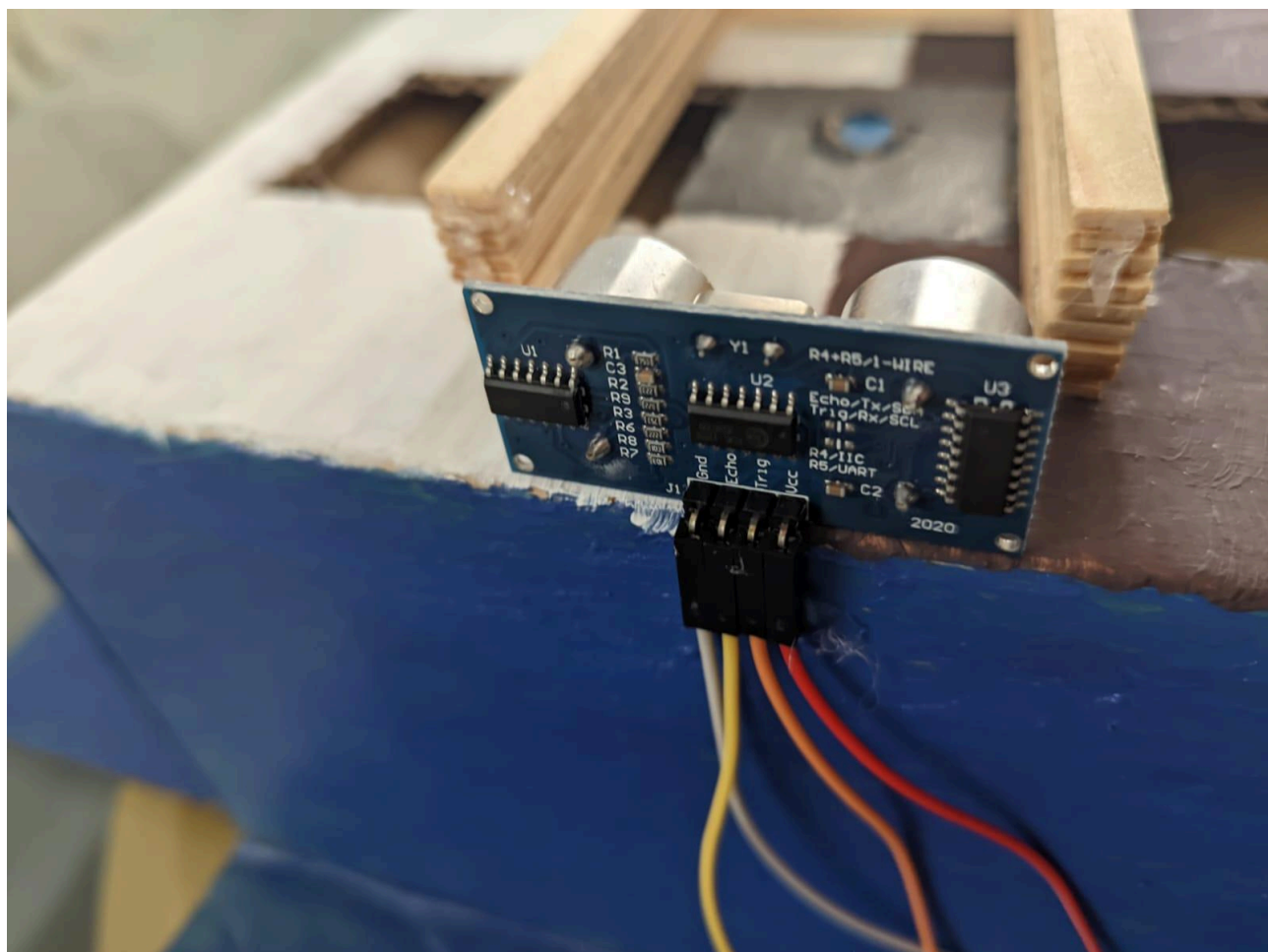
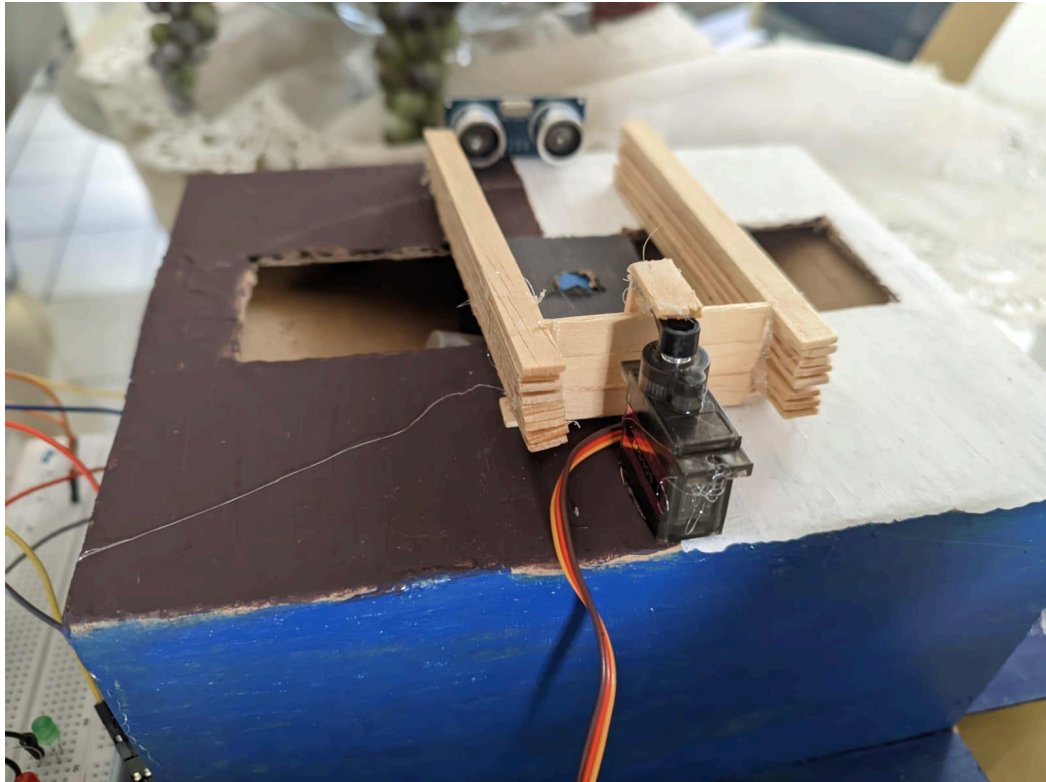


Figura 8 – Esquema de posicionamento do Servo Motor



Demonstração do Projeto:

<https://youtu.be/1V27HVFb9t4>

Slides de Apresentação do Projeto:

https://www.canva.com/design/DAGhEU_5RfM/VIrNGI6QOUL31uaBE6Jspg/view?utm_content=DAGhEU_5RfM&utm_campaign=designshare&utm_medium=link2&utm_source=uniquelinks&utlId=h2184d8f2f3

Código Utilizado:

```
#define F_CPU 16000000UL
#include <Servo.h>
#include <Ultrasonic.h>
#include <avr/io.h>
#include <avr/interrupt.h>
```

Servo meuServo;

Ultrasonic sensorUltrassonico(0b100, 0b101); // Pinos 4 (TRIG) e 5 (ECHO)


```

// Variáveis Globais

volatile bool detectouMetal = false;
volatile int contador2s = 0; // Contador para 2 segundos
volatile int contador4s = 0; // Contador para 4 segundos
volatile int contadorMovServo = 0; // Contador para movimento do servo
bool objetoDetectado = false;
bool movimentoServo = false;

void setup() {
    // Configuração de portas (PD4=TRIG como saída, PD2=entrada, PD6/PD7=LEDs)
    DDRD = 0b11010000; // PD4(TRIG), PD6 e PD7 como saídas
    PORTD = 0b00000100; // Pull-up em PD2

    Serial.begin(9600);
    meuServo.attach(0b1001); // Servo no pino 9
    meuServo.write(90);

    // Configuração de interrupções
    EICRA |= (1 << ISC01); // INT0 borda de descida
    EIMSK |= (1 << INT0); // Habilita INT0

    // Configura Timer2 para 1ms (prescaler 64)
    cli();
    TCCR2B |= (1 << CS22); // Prescaler 64
    TIMSK2 |= (1 << TOIE2); // Habilita overflow
    TCNT2 = 5; // Inicializa contador
    sei();
}

void loop() {
    if (!objetoDetectado) {
        long microsec = sensorUltrassonico.timing();
        float distancia = sensorUltrassonico.convert(microsec, Ultrasonic::CM);

        Serial.print("Distância: ");
    }
}

```

```

Serial.print(distancia);
Serial.println(" cm");

if (distancia > 0 && distancia <= 8) {
    objetoDetectado = true;
    contador2s = 0;
    contador4s = 0;
    Serial.println("Objeto detectado, aguardando sensor de metal...");
}
}

if (objetoDetectado && !movimentoServo) {
    int cont2s, cont4s;
    cli();
    cont2s = contador2s;
    cont4s = contador4s;
    sei();

    if (detectouMetal && cont2s >= 2000) { // 2000 ms = 2 segundos
        Serial.println("Metal detectado! Movendo para direita...");
        PORTD = 0b01000000; // LED verde
        meuServo.write(45);
        contadorMovServo = 0; // Inicia contagem para retorno
        movimentoServo = true;
        detectouMetal = false;
    }
    else if (cont4s >= 4000) { // 4000 ms = 4 segundos
        Serial.println("Nenhum metal detectado após 4s. Movendo para esquerda...");
        PORTD = 0b10000000; // LED vermelho
        meuServo.write(135);
        contadorMovServo = 0; // Inicia contagem para retorno
        movimentoServo = true;
    }
}
}

```

```

if (movimentoServo) {
    int contMovServo;
    cli();
    contMovServo = contadorMovServo;
    sei();

    if (contMovServo >= 700) { // 700 ms para retornar ao centro
        meuServo.write(90);
        PORTD = 0b00000000; // LEDs off
        objetoDetectado = false;
        movimentoServo = false;
        contadorMovServo = 0; // Reset
        Serial.println("Retornando ao centro - Descarte realizado!");
    }
}
}
}

```

```

// Interrupção do sensor de metal
ISR(INT0_vect) {
    detectouMetal = true;
    Serial.println("Metal detectado!");
}

```

```

// Interrupção do Timer2 (1ms)
ISR(TIMER2_OVF_vect) {
    TCNT2 = 5; // Reinicia contador

```

```

    if (objetoDetectado) {
        contador2s++;
        contador4s++;
    }

```

```

    if (movimentoServo) {
        contadorMovServo++;
    }

```


}

5. Conclusões

O sistema desenvolvido foi capaz de detectar objetos e materiais metálicos e não metálicos de forma eficiente, acionando o servo motor conforme a necessidade solicitada no envio(trig) na detecção e envio de sinal(Echo). Os resultados práticos obtidos durante os testes estavam em conformidade com os cálculos teóricos e os resultados da simulação. O uso de interrupções externas para o sensor de metal e a temporização da interrupção interna do movimento do servo motor garantiram uma operação fluida e precisa. A montagem experimental demonstrou que o sistema atende aos requisitos de automação propostos, com boa precisão na detecção e controle do movimento para separação de material, podendo ser ajustada a escala de tempo de espera baseado na sua necessidade, onde nos testes o tempo utilizado para metal foi 2 segundos e 4 segundos para não metal.

O erro percentual nas comparações entre os resultados simulados e os experimentais foi inferior a 5%, onde durante os testes práticos foi perceptível que o número de prescale junto ao TCNT correto influência diretamente no tempo de resposta, pois ao escolher valores que geram números decimais no cálculo, gera resultados imprecisos no movimento do servo, já que o tempo precisa ser um número sem casas decimais para cada 1ms, então eu considero como uma das partes mais fundamentais essa sincronização do timer para obtenção correta dos movimentos no tempo de espera entre o ciclo de cada movimento.

Referências Bibliográficas:

- <https://www.youtube.com/watch?v=jvR-AffZ6jQ&t=15s>
- <https://www.youtube.com/watch?v=ZjmbILOc6Vk>
- <https://www.makerhero.com/blog/o-que-e-servo-motor/>
- <https://www.makerhero.com/blog/controlando-um-servo-motor-com-arduino/>
- https://www.baumer.com/br/pt/service-support/modo-de-funcionamento/funcionamento-de-sensores-ultrassonicos/a/Know-how_Function_Ultrasonic-sensors
- <https://www.casadarobotica.com/sensores-e-modulos/sensores/movimento-e-proximidade/sensor-indutivo-npn-de-proximidade-lj12a3-4-z-bx>
- https://github.com/marceloifpe/PROJETO-SISTEMA_DE_SEPARACAO_METAL_E_NAO_METAL
- <https://cv.antonello.com.br/robo-com-sensor-ultrasonico/>