



**CENTRO UNIVERSITÁRIO MÁRIO PONTES JUCÁ**  
**CURSO DE ENGENHARIA DE COMPUTAÇÃO**

## **RELATÓRIO DA EXPERIÊNCIA Nº1**

### **BALL AND BEAM**

Turma 2023.1

Turma 2023.1

Alagoas - AL



Maceió - AL

## **BALL AND BEAM**

Primeiro Relatório Parcial apresentado à disciplina de Projeto de Automação e Controle, correspondente à avaliação do semestre 2023.1 do 9º período do curso de Engenharia de Computação da Centro Universitário Mário Pontes Jucá, sob orientação do **Prof. Luís Felipe Vieira Silva**.

Turma 2023.1

Alagoas - AL

## **RESUMO**

Com o avanço tecnológico, tornou-se essencial encontrar uma solução para automatizar tarefas simples e repetitivas com precisão e segurança. Nesse contexto, a utilização do controlador PID surge como uma solução para atender essa crescente necessidade. No presente trabalho, iremos empregar o controlador PID em um sistema ball and beam. No decorrer deste trabalho será demonstrado como fazer a modelagem do sistema, sua construção e funcionamento.

**Palavras-chave: Ball and beam, PID.**

## LISTA DE FIGURAS

<b>Figura 1.</b>	ARDUINO UNO.....	8
<b>Figura 2.</b>	SERVO MOTOR.....	9
<b>Figura 3.</b>	PROTOBOARD.....	9
<b>Figura 4.</b>	SENSOR ULTRASSÔNICO.....	10
<b>Figura 5.</b>	JUMPER.....	11
<b>Figura 6.</b>	BOLA.....	12
<b>Figura 7.</b>	GRÁFICO PROPORCIONAL.....	13
<b>Figura 8.</b>	GRÁFICO INTEGRAL.....	13
<b>Figura 9.</b>	GRÁFICO DERIVATIVO.....	14
<b>Figura 10.</b>	ESQUEMÁTICA BALL AND BEAM.....	15
<b>Figura 11.</b>	PROJETO BALL AND BEAM.....	16
<b>Figura 12.</b>	DIAGRAMA DE BLOCOS.....	17
<b>Figura 13.</b>	CÓDIGO PARTE 1.....	18
<b>Figura 14.</b>	CÓDIGO PARTE 2.....	19
<b>Figura 15.</b>	CÓDIGO PARTE 3.....	19
<b>Figura 16.</b>	BARRA PRINCIPAL ATIVADA.....	20
<b>Figura 17.</b>	BARRA PRINCIPAL DESATIVADA.....	21

## SUMÁRIO

	<u>Pág.</u>
<b>1 INTRODUÇÃO</b>	<b>6</b>
<b>2 REFERENCIAL TEÓRICO</b>	<b>7</b>
2.1 Sistema Ball and beam	7
2.2 Componentes e diagrama do sensor ball and beam	8
2.2.1 Arduino Uno	8
2.2.2 Servo motor	8
2.2.3 Protoboard	9
2.2.4 Sensor ultrassônico	10
2.2.5 Jumper	10
2.2.6 Barra e bola	11
2.2.7 Controlador PID	15
<b>3 METODOLOGIA</b>	<b>16</b>
3.1 Estrutura ball and beam	16
3.2 Diagrama de blocos do sistema	17
3.3 Código utilizado para controle do sistema	18
<b>4 RESULTADOS</b>	<b>20</b>
<b>5 CONCLUSÃO</b>	<b>22</b>
<b>6 REFERÊNCIAS</b>	<b>13</b>
<b>7 ANEXOS</b>	<b>14</b>
<b>8 APÊNDICES</b>	<b>15</b>

# 1 INTRODUÇÃO

Sistemas de controle são uma presença constante e vital dentro da sociedade moderna. Esses sistemas desempenham papel crucial na automação e gerenciamento eficiente de processos em diversos setores, desde a indústria até os serviços públicos. Tais sistemas são projetados para monitorar e regular o desempenho de sistemas complexos, garantindo que eles operem dentro dos limites aceitáveis e alcancem os objetivos desejados. Esses sistemas desempenham um papel essencial no aprimoramento da eficiência, produtividade e segurança em uma ampla gama de aplicações.

No contexto da sociedade, os sistemas de controle podem ser encontrados em muitos aspectos da vida cotidiana, por exemplo, o sistema de controle de tráfego gerencia o fluxo de veículos nas ruas, otimizando a distribuição do tráfego e reduzindo o congestionamento. Como afirmou o engenheiro de controle Karl Johan Åström, "Os sistemas de controle de tráfego são vitais para melhorar a eficiência do transporte e reduzir o consumo de combustível e as emissões de poluentes."

Dentro do contexto dos sistemas de controle abordados anteriormente, um exemplo particularmente interessante é o sistema Ball and Beam, assunto que será abordado neste relatório. Esse sistema ilustra de forma prática os princípios e aplicações dos sistemas de controle na estabilização e controle de um objeto em movimento.

## **2 REFERENCIAL TEÓRICO**

### **2.1 Sistema Ball and Beam**

O sistema Ball and Beam é um exemplo prático de como os princípios de controle podem ser aplicados para estabilizar um objeto em movimento. Ele utiliza um controlador PID para ajustar a posição da esfera em cima de uma viga. O servo motor é o componente responsável por mover a viga conforme as correções do controlador PID. Além disso, o Arduino, uma plataforma de prototipagem eletrônica, é frequentemente utilizado para controlar o sistema Ball and Beam. Com o Arduino é possível programar e controlar o servo motor e o controlador PID, permitindo um controle preciso e automático do sistema. Essa combinação de elementos Ball and Beam, controle PID, servo motor e Arduino exemplifica como a integração de diferentes tecnologias pode criar soluções avançadas de controle e automação em diversos campos.

## 2.2 Componentes e diagrama do Sistema Ball and Beam

### 2.2.1 Arduino Uno

Foi utilizado o microcontrolador ATMEGA 281P localizado no arduino uno com um clock de 16MHz, 14 pinos de entrada ou saída, 6 entradas analógicas, uma entrada de energia, um conector usb e um botão de reset. Ele possui todo o necessário para suportar o microcontrolador.



Figura 1: Arduino Uno

Fonte: Components 101

### 2.2.2 Servo Motor

Para a movimentação da viga foi utilizado um servo motor, que é usado em várias aplicações quando se deseja movimentar algo de forma precisa e controlada. Sua característica mais marcante é a sua capacidade de movimentar os seu braço até uma posição e mantê-lo, mesmo quando sofre uma força em outra direção.



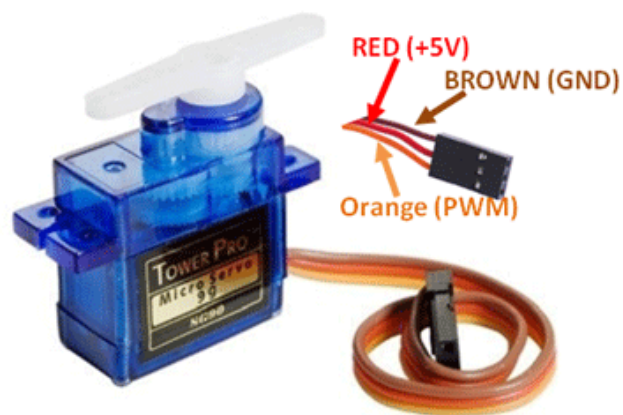


Figura 2: Servo Motor

Fonte: components 101

### 2.2.3 Protoboard

Protoboard é uma placa de ensaio que serve como um protótipo de um aparelho eletrônico, com uma matriz de contatos que possibilita construir circuitos de teste sem que haja necessidade de solda e, assim, garantindo segurança e agilidade em diferentes atividades.

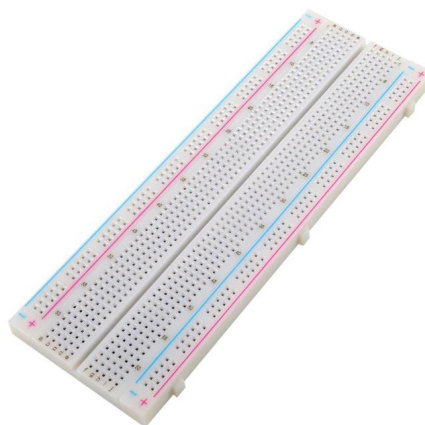


Figura 3: Protoboard

Fonte: Curto Circuito

### 2.2.4 Sensor ultrassônico

Um sensor ultrassônico é um dispositivo eletrônico que utiliza ondas sonoras de alta frequência, além do alcance da audição humana (geralmente acima de 20 kHz), para detectar a presença de objetos e medir distâncias. Esse tipo de sensor emite pulsos ultrassônicos e mede o tempo que leva para os pulsos retornarem após atingirem um objeto. Assim, o sensor pode determinar a distância até o objeto. Os sensores ultrassônicos são frequentemente utilizados para detecção de obstáculos, navegação e medição de distâncias em uma ampla variedade de aplicações.

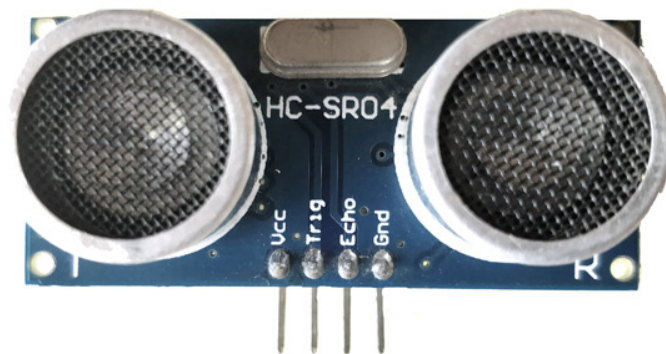


Figura 4: Sensor ultrassônico

Fonte: Components 101

### 2.2.5 Jumper

É um dispositivo utilizado em eletrônica e informática para realizar conexões elétricas temporárias ou alterar a configuração de um circuito. Geralmente, um jumper consiste em um pequeno conector metálico que pode ser inserido em um conjunto de pinos ou terminais em um dispositivo ou placa de circuito impresso.



Figura 5: Jumper

Fonte: Eletrogate

### 2.2.6 Barra e Bola

O sistema é composto por uma barra horizontal e uma bola que pode rolar livremente ao longo dessa barra. O objetivo é controlar a posição da bola na barra, mantendo-a em uma posição desejada ou movendo-a para uma posição específica.



Figura 6: Bola rollon

Fonte: Magalu

### 2.2.7 Controlador PID

Trata-se de um algoritmo empregado em sistemas de controle automático, tais como motores ou quaisquer outros sistemas destinados à automação. Esse algoritmo pode ser subdividido em três componentes distintos:

**P - Proporcional:** Ele é calculado ao multiplicarmos o erro atual pela constante de ganho, como o próprio nome se refere, proporcional. Em outras palavras, por se tratar de um algoritmo que é utilizado para automação em tempo real, esta parte seria a responsável por começar a mudança do valor atual para o valor desejado, porém se apenas esta parte do algoritmo for utilizada o valor final irá apresentar um erro, como está representado no gráfico abaixo:

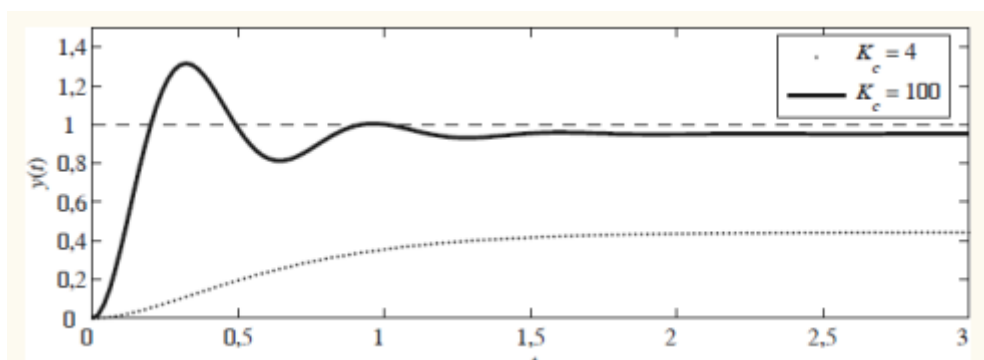


Figura 7: Gráfico proporcional

Fonte: material de apoio fornecido pelo orientador

**I - Integral:** Calculado a partir da integração do erro anterior com a multiplicação da constante de ganho integral. Esta parte do algoritmo será responsável por retirar o erro do resultado final, porém, isso acarretará uma lentidão até alcançar o resultado desejado, como está representado no gráfico abaixo:

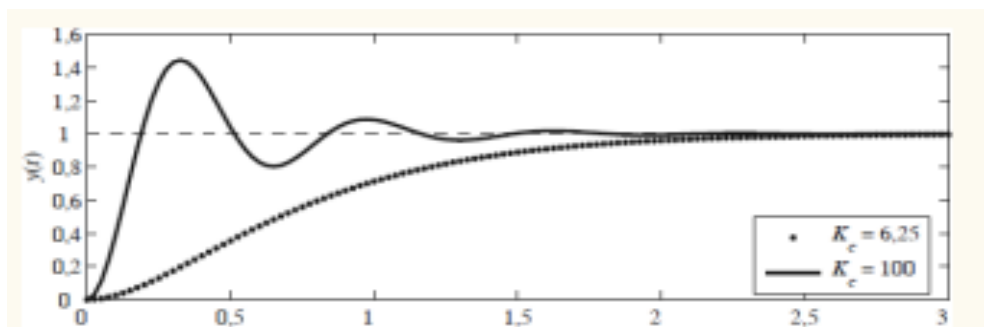


Figura 8: Gráfico integral

Fonte: material de apoio fornecido pelo orientador

D - Derivativo: Calculado derivando o erro em relação ao tempo e multiplicando pela constante de ganho derivado. Esta última parte do algoritmo é responsável por reverter o efeito da parte integral, onde ele será responsável por obter de forma acelerada os resultados desejados, como está representado no gráfico abaixo:

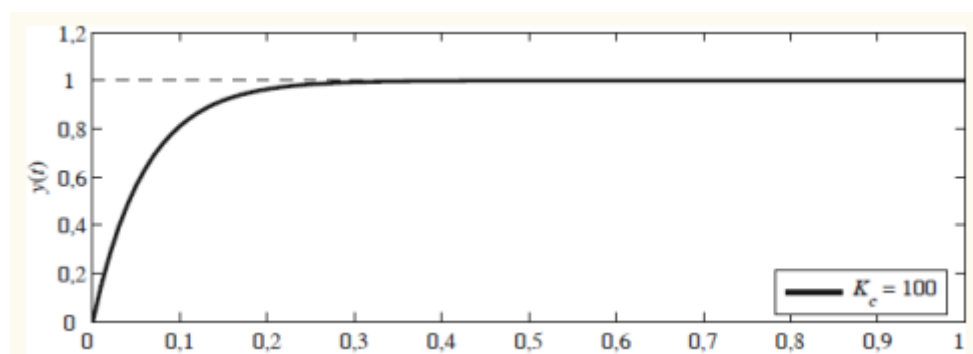


Figura 9: Gráfico derivativo

Fonte: material de apoio fornecido pelo orientador

### 3 METODOLOGIA

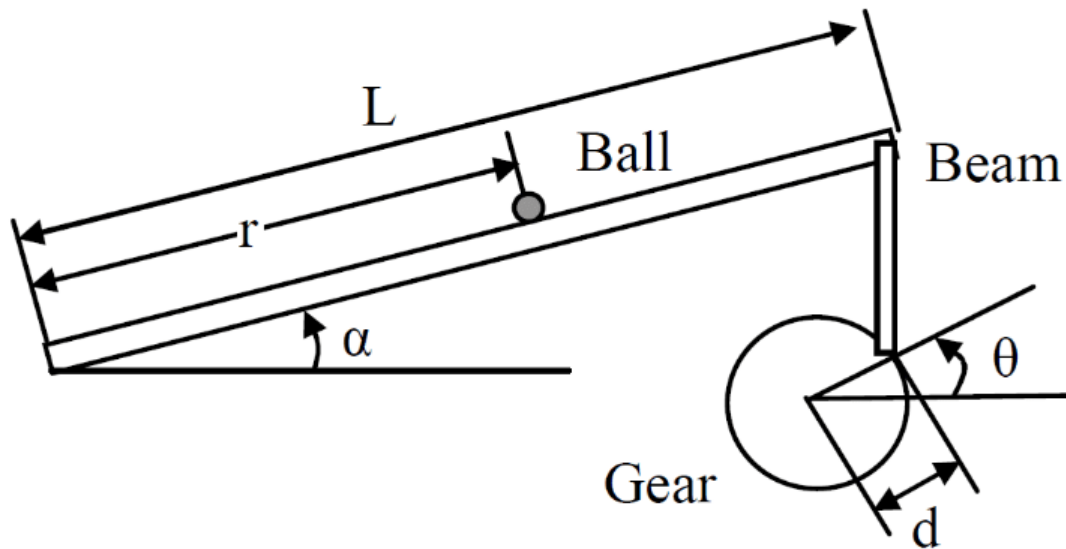


Figura 10: Esquemática ball and beam

A figura acima demonstra o esquemático base para a montagem de um projeto ball and beam, onde ele é composto por uma base, uma barra principal, um servo motor, uma outra barra para servir de alavanca conectando o servo motor e a barra, um suporte para a barra e uma bola.

A barra principal é então fixada de um dos lados no suporte e no outro na outra barra servindo de alavanca, o qual irá movimentar a barra de acordo com a rotação do motor.

### 3.1 Estrutura ball and beam

A imagem e vídeo abaixo irão demonstrar o visual final do sistema após a montagem e teste bem como sua funcionalização.

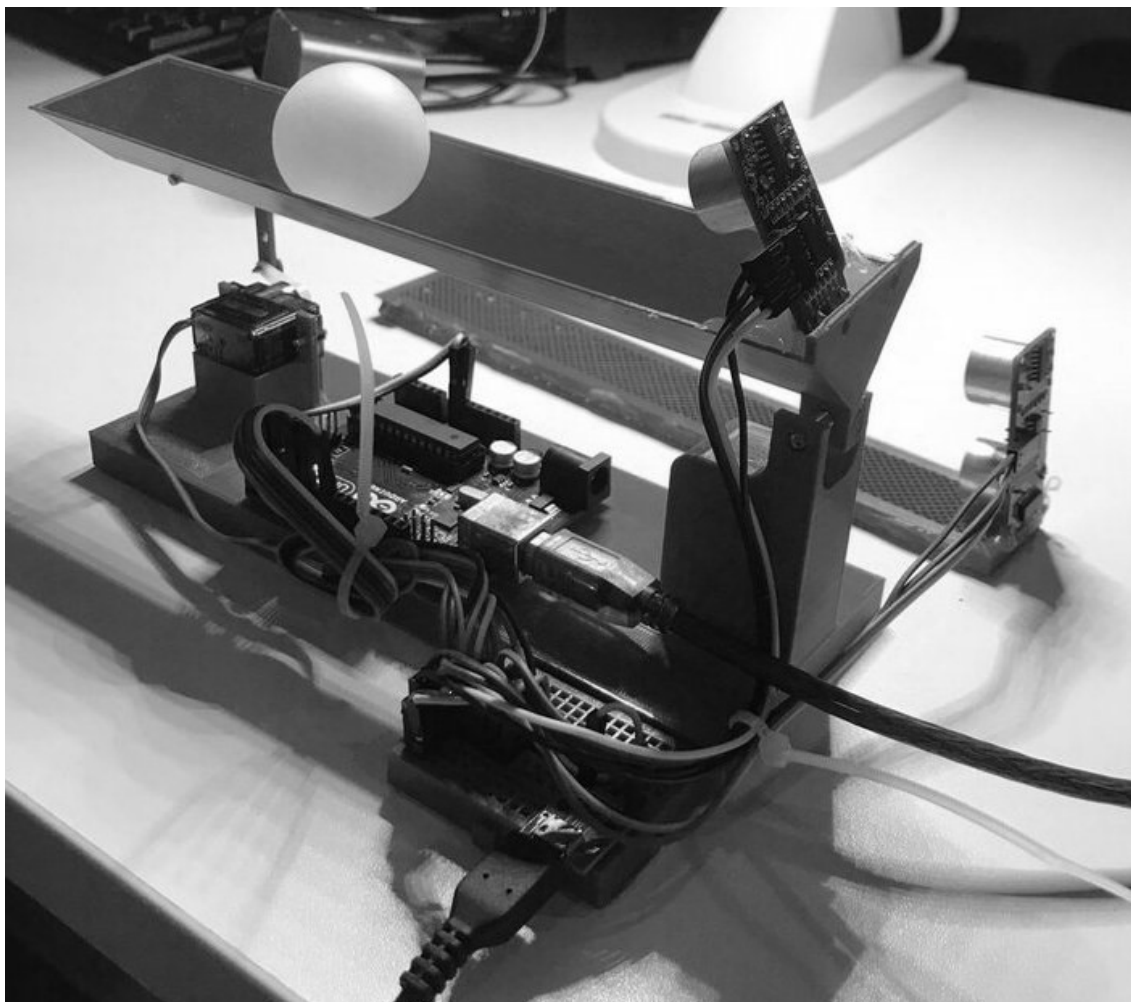


Figura 11: Projeto ball and beam

Fonte: Feito pelos autores do trabalho

[Video do projeto em execução](#)

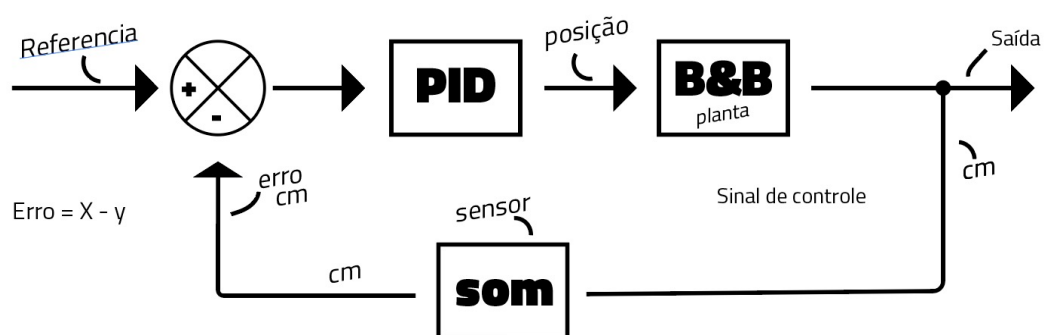


Turma 2023.1

Alagoas - AL

### 3.2 Diagrama de blocos do sistema

A imagem abaixo irá demonstrar o visual do esquemático do diagrama de blocos baseado na construção do projeto.



Fonte: Feito pelo próprio autor

Figura 12: Diagrama de blocos



### 3.3 Código utilizado para o controle do sistema

[Link do código no github](#)

```
127 lines (92 sloc) | 2.4 KB
1 #include <Servo.h>
2
3 #define UMAX_DEG 66
4 #define UMIN_DEG -66
5 #define UMAX_RAD 1.151
6 #define UMIN_RAD -1.151
7 #define T 0.09
8
9 const int echoPin1 = 6;
10 const int trigPin1 = 7;
11
12 Servo servo;
13
14 double setpoint, setpoint_prec; // Em metros: 30cm --> 0.3m
15 double y, y_prec;
16 double error;
17 double P, I, D, U;
18 double I_prec = 0, U_prec = 0, D_prec = 0;
19 boolean saturation = false;
20
21 double Kp = 5.8;
22 double Ki = 1.3;
23 double Kd = 2.5;
24
25 float measure_1(void);
26 void move_servo(int);
27
28 void setup() {
29     Serial.begin(9600);
30
31     pinMode(trigPin1, OUTPUT);
32     pinMode(echoPin1, INPUT);
33     servo.attach(9);
34
35     delay(1000);
36     move_servo(90);
37     delay(2000);
38     setpoint_prec = 0.09;
39     delay(1000);
40     y_prec = measure_1(); // carrinho
41     delay(1000);
42 }
43
44 void loop() {
45     setpoint = 0.09;
46 }
```

Figura 13: Código parte 1

```

47   delay(3);
48
49   y = measure_1();
50
51   delay(3);
52
53   error = round(100 * (y - setpoint)) * 0.01; // Em metros
54
55   P = Kp * error;
56
57   if (!saturation)
58     I = I_prec + T * Ki * error;
59
60   D = (Kd / T) * (y - y_prec);
61
62   D = 0.56 * D + 0.44 * D_prec;
63
64   U = P + I + round(100 * D) * 0.01; // U em radianos
65
66   if (U < UMIN_RAD) {
67     U = UMIN_RAD;
68     saturation = true;
69   } else if (U > UMAX_RAD) {
70     U = UMAX_RAD;
71     saturation = true;
72   } else {
73     saturation = false;
74   }
75
76   U = round(U * 180 / M_PI); // Transforma U em graus: -63° < U < 63°
77
78   U = map(U, UMIN_DEG, UMAX_DEG, 24, 156); // Mapeia o valor calculado de U para o valor correspondente do servo
79
80   if (U < 83 || U > 95 || abs(error) > 0.02) {
81     move_servo(round(U));
82   }
83
84   delay(24);
85
86   Serial.print(setpoint * 100);
87   Serial.print(" ");
88   Serial.print(y * 100);
89   Serial.print(" ");
90   Serial.print(U);
91   Serial.println();
92
93   I_prec = I;
94   y_prec = y;

```

Figura 14: Código parte 2

```

95  D_prec = 0;
96  setpoint_prec = setpoint;
97  }
98
99  float measure_1(void) {
100    long duration = 0;
101    float distance = 0;
102
103    digitalWrite(trigPin1, LOW);
104    delayMicroseconds(10);
105
106    digitalWrite(trigPin1, HIGH);
107    delayMicroseconds(10);
108
109    digitalWrite(trigPin1, LOW);
110
111    duration = pulseIn(echoPin1, HIGH);
112    distance = (float)duration / 58.2;
113
114    delay(30);
115
116    if (distance > 20) {
117      distance = 20;
118    } else if (distance < 0) {
119      distance = 0;
120    }
121
122    return 0.01 * (distance - 1.5 + 0.5); // metros
123  }
124
125  void move_servo(int u) {
126    servo.write(u - map(u, 30, 150, 14, 3));
127  }

```

Figura 14: Código parte 3

## 4 RESULTADOS

Durante o procedimento de prototipagem do projeto proposto, foi posto em prática os conteúdos vistos em sala de aula a fim de observar a relação teoria-prática em ação. Foram enfrentados desafios durante o processo para a realização do projeto, como por exemplo, o material da esfera inicial que era feita de um material permeável que ocasionava na absorção das ondas do sensor ultrassônico, o que resultava em uma má leitura do sensor, a situação foi contornada ao ser utilizada uma esfera feita de outro material que não possibilita a absorção de ondas sonoras, desta forma foi obtido êxito no projeto.

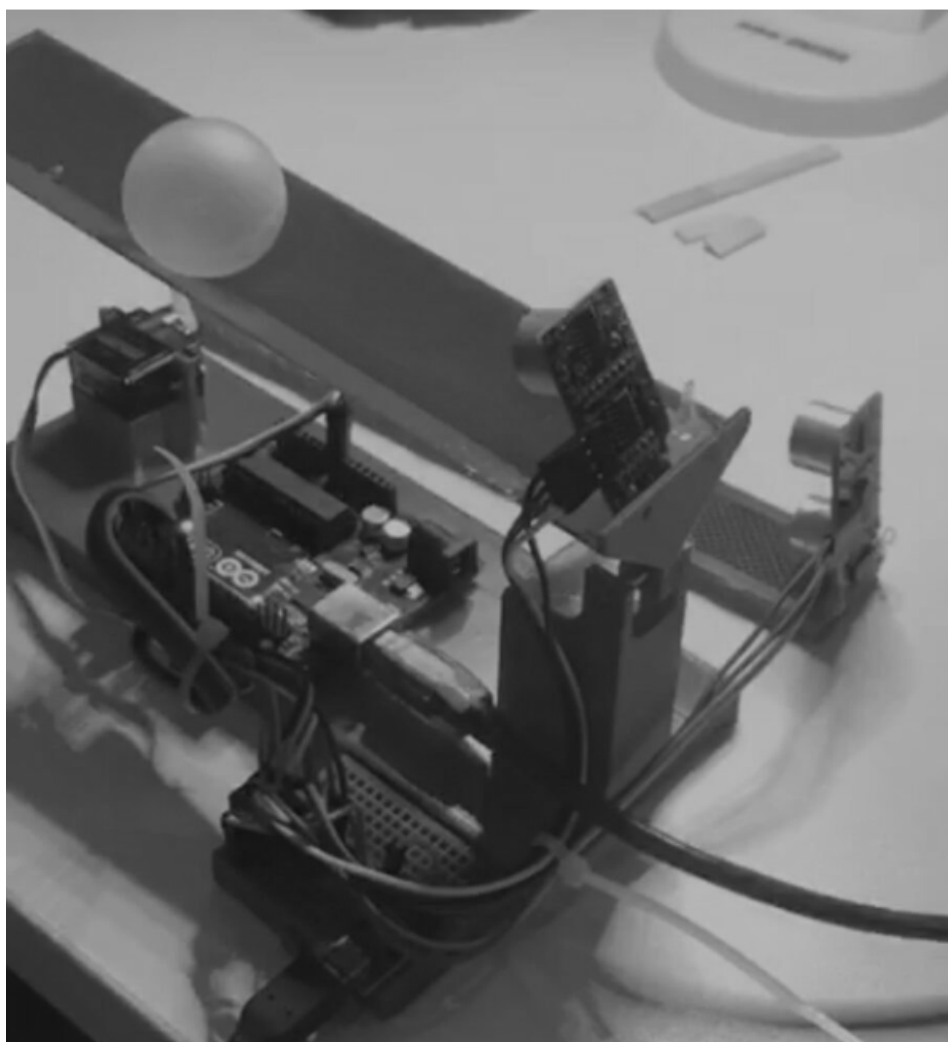


Figura 16: Barra principal ativada

Fonte - Próprio autor

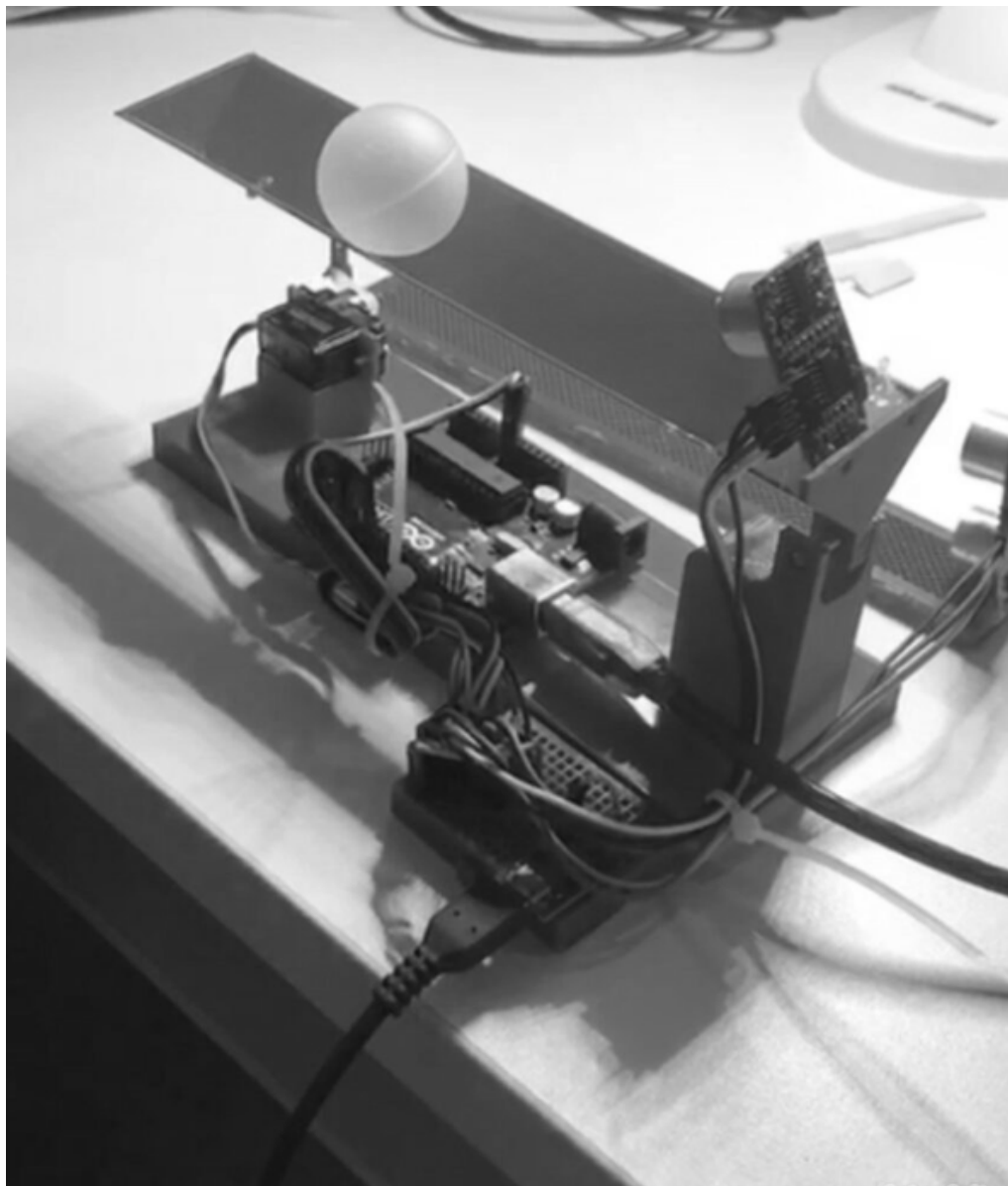


Figura 17: Barra principal desativada

Fonte: O próprio autor

Turma 2023.1

Alagoas - AL

## **5 CONCLUSÃO**

Durante o processo de prototipagem do projeto, identificou-se uma dificuldade relacionada à posição do sensor. Ao ser instalado em uma altura inferior à da barra principal, verificou-se que, durante sua ativação, o sensor poderia deixar de detectar a bolinha. Diante dessa situação, optou-se por elevar a posição do sensor a fim de solucionar o problema.

Durante o restante da implementação do projeto, os componentes operaram de maneira impecável, em conformidade com as expectativas estabelecidas.

Por meio da conclusão do projeto, foi possível alcançar uma solução de baixo custo e altamente prática para abordar problemas que envolvem tarefas repetitivas e requerem automação. Além disso, obteve-se resultados precisos e a capacidade de escalonamento para atender às necessidades das indústrias.

## REFERÊNCIAS

Nise, N. S. **Engenharia de Sistemas de Controle**. 6.ed. California State Polytechnic University, Pomona

Nise, N. S. **Engenharia de Sistemas de Controle**. 7.ed. California State Polytechnic University, Pomona

**Protoboard 830 Pontos**. Disponível em:

<<https://curtocircuito.com.br/protoboard-830-pontos.html>>. Acesso em: 13 jun. 2023.

**Servo motor SG - 90. Components 101, 2017**. Disponível em:

<<https://components101.com/motors/servo-motor-basics-pinout-datasheet/>>. Acesso em: 13 de jun. de 2023.

**Arduino Uno. Components 101, 2021**. Disponível em:

<<https://components101.com/microcontrollers/arduino-uno/>>. Acesso em: 13 de jun. de 2023.

**HC-SR04 Ultrasonic sensor. Components 101, 2021**. Disponível em:

<<https://components101.com/sensors/ultrasonic-sensor-working-pinout-datasheet/>>. Acesso em: 13 de jun. de 2023.

**Protoboard 830 Pontos**. Disponível em:

<<https://curtocircuito.com.br/protoboard-830-pontos.html>>. Acesso em: 13 de jun. de 2023.

**Jumpers - Macho/Macho**. Disponível em:

<<https://www.eletrogate.com/jumpers-macho-macho-40-unidades-de-10-cm>>. Acesso em: 13 de jun. de 2023.

Github do projeto:

<<https://github.com/andreyjackson/Projeto-de-Automacao-e-Control-PID/>>.