

## Sumário

1. Objetivo .....	3
2. Resumo.....	3
3. Procedimento de desenvolvimento.....	3
3.1 Materiais e equipamentos utilizados .....	3
3.2 Projeto.....	3
4. Resultados e Discussão .....	10
5. Conclusão .....	11
6. Referências Bibliográficas .....	11

## 1. Objetivo

Esse trabalho tem como objetivo construir um encoder absoluto. Com intuito de entender noções de eletrônica e lógica de código de decodificação de posição angular do encoder (código gray ou binário), a partir do desenvolvimento dessa implementação na prática. Afim de construir um encoder absoluto capaz de identificar a posição angular proveniente da rotação do eixo de motor.

## 2. Resumo

Esse trabalho apresenta uma implementação de encoder absoluto. Contendo o projeto de desenvolvimento desse encoder e os materiais utilizados.

## 3. Procedimento de desenvolvimento

### 3.1 Matérias e Equipamentos utilizados

- Micro Motor DC N20 Com Redução - 6V 200RPM;
- 5 LEDs Infravermelho 3mm Emissor, e 5 fototransistores;
- Placa de desenvolvimento Arduino;
- Fonte de bancada;
- Porcas, parafusos e espaçadores metálicos;
- Filamentos de impressora 3D (PLA e ABS);
- Impressora 3D (para imprimir o chassi);
- Jumpers;
- Ferro de solda e estanho;
- Tubo termo retrátil;
- Software SketchUp;
- Software Simplify3D;
- Software Arduino IDE.

### 3.2 Projeto

#### **Estrutura do encoder absoluto**

Esse projeto de encoder foi desenvolvido em primeiro momento a modelagem do encoder absoluto (estrutura para fixação dos sensores e base do encoder, suporte do motor

e o disco com código gray) em um software 3D (Figuras 1, 2 e 3) chamado SketchUp. Com a modelagem feita, foi gerado o arquivo STL (contém dados que descrevem o layout de um objeto tridimensional) das peças. Após isso foi utilizado um fatiador de impressão 3D chamado Simplify3D, em que configurei o tamanho e densidade para otimizar uso de material e tempo de impressão das peças, com isso configurado gerei o G-code (instrução de máquina das peças 3D) para imprimir as peças em uma impressora 3D.

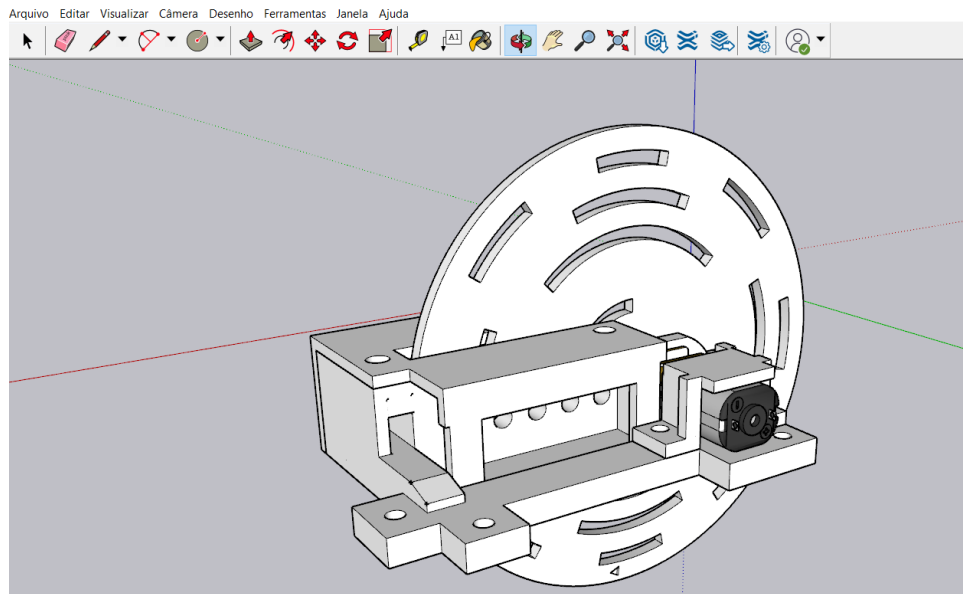


Figura 1 – Modelagem estrutura encoder absoluto (vista 1).

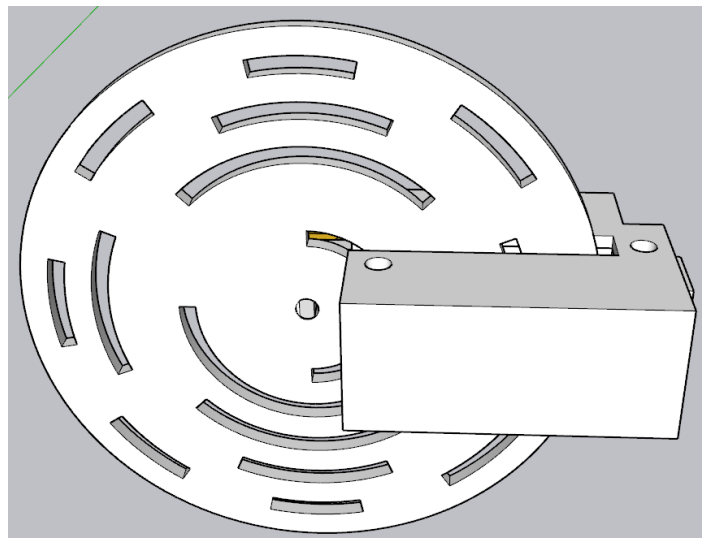


Figura 2 – Modelagem estrutura encoder absoluto (vista 2).

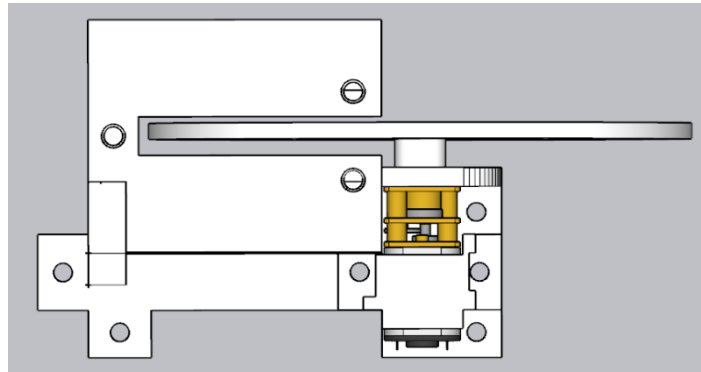


Figura 3 – Modelagem estrutura encoder absoluto (vista 3).

O disco do encoder absoluto, foi projetado para ter 5 bits de combinação baseada no código Gray em que o código é não ponderado onde de um número para outro apenas um bit varia, devido a essa variação de apenas um bit por vez garante na leitura dos sensores haver menos erros de leitura, diferente do código binário que a taxa de variação de bits é maior. O disco está representando pela Figura 4 e suas possíveis combinações na Tabela 1.

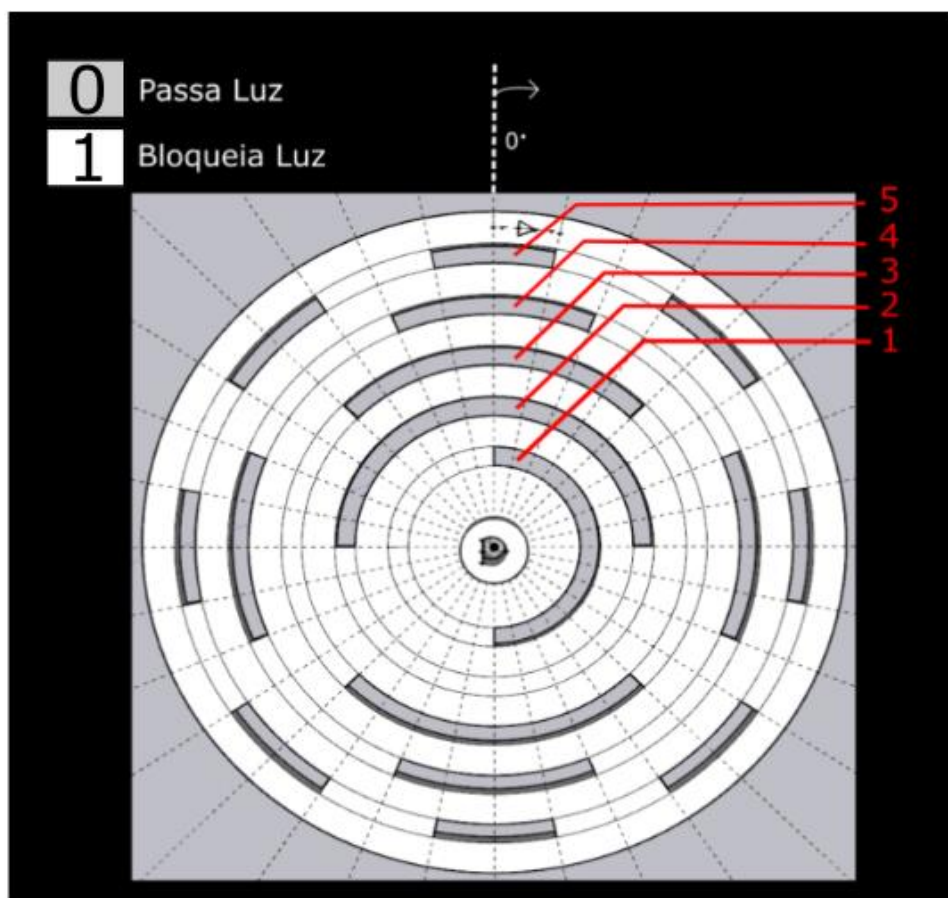


Figura 4 – Codificador rotativo absoluto com 5 linhas de fendas

**Tabela 1: Combinações possíveis do encoder absoluto de 5 bits com código Gray:**

ÂNGULO (°)	1	2	3	4	5
0,00	0	0	0	0	0
11,25	0	0	0	0	1
22,50	0	0	0	1	1
33,75	0	0	0	1	0
45,00	0	0	1	1	0
56,25	0	0	1	1	1
67,50	0	0	1	0	1
78,75	0	0	1	0	0
90,00	0	1	1	0	0
101,25	0	1	1	0	1
112,25	0	1	1	1	1
123,75	0	1	1	1	0
135,00	0	1	0	1	0
146,25	0	1	0	1	1
157,50	0	1	0	0	1
168,75	0	1	0	0	0
180,00	1	1	0	0	0
191,25	1	1	0	0	1
202,50	1	1	0	1	1
213,75	1	1	0	1	0
225,00	1	1	1	1	0
236,25	1	1	1	1	1
247,50	1	1	1	0	1
258,75	1	1	1	0	0
270,00	1	0	1	0	0
281,25	1	0	1	0	1
292,50	1	0	1	1	1
303,75	1	0	1	1	0
315,00	1	0	0	1	0
326,25	1	0	0	1	1
337,50	1	0	0	0	1
348,75	1	0	0	0	0

### **Eletrônica**

Referente a eletrônica de sensores, foi feito utilizando 5 LEDs infravermelhos emissores e 5 fototransistores, a ligação elétrica foi feita conforme o esquemático eletrônico na Figura 5. Foram utilizados resistores (R1, R2, R3, R4 e R5), para limitação de corrente dos LEDs infravermelhos (D1, D2, D3, D4 e D5) e resistores de Pull-Up (R6, R7, R8, R9 e R10) nos coletores dos transistores (Q1, Q2, Q3, Q4 e Q5) para obter um nível de tensão variável (S1, S2, S3, S4 e S5) conforme a intensidade luminosa no fototransistor proveniente da emissão de infravermelho do led que provocará a mudança de tensão nos coletores dos transistores.

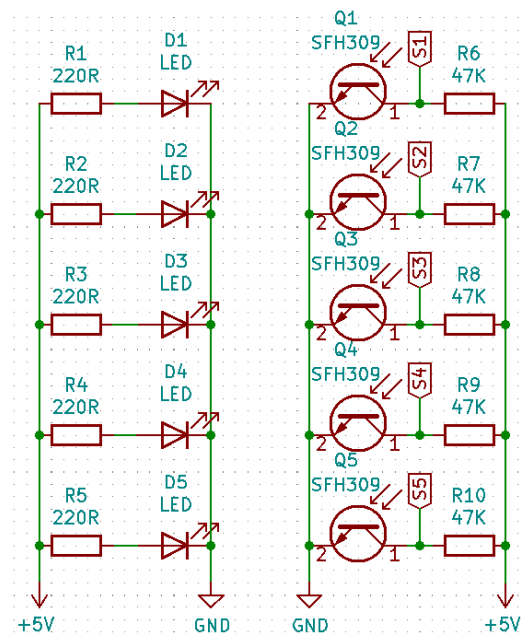


Figura 5 – Esquema eletrônico dos sensores infravermelhos.

O circuito desse sensor foi desenvolvido no software KiCad, em que foi utilizado para fazer o esquema eletrônico do circuito.

Para fazer a aquisição do sinal proveniente desses sensores, foi utilizado uma placa de desenvolvimento embarcada chamada Arduino (Figura 6). Em que foi desenvolvido um código para leitura dos sensores que será abordado na secção de código.



Figura 6 – Plataforma de desenvolvimento embarcado Arduino.

### **Motores e rodas**

Para a rotação do disco do encoder foi utilizado um motor (Figura 7). O modelo utilizado é um motor de 300 rpm e com caixa de redução que permite um torque de:

- Torque nominal: 0,3 Kg.cm;
- Torque de parada: 1,8 Kg.cm;



Figura 7 - Motor.

### **Código**

Para leitura dos sensores e decodificação do código Gray do disco do encoder, foi feito um código utilizando a IDE do arduino para isso. Nesse código ele faz a leitura de cada sensor, obtendo um vetor de 5 bits sendo 1 bit representando cada sensor, e nesse vetor de 5 bits contém um código Gray (o disco do encoder possui buracos em umas partes e fechados em outras, que permitem a passagem de luz (sensor lê 0 V ou 0) e não permitem a passagem de luz (o sensor lê 5 V ou 1), entendendo esse padrão do código

Gray é possível identificar a posição angular do eixo do motor pois o disco tem 360 graus e 5 bits (32 posições possíveis). Abaixo será apresentado o código principal:

### Código principal:

```
1 boolean pin_state[5];
2 byte input_pin[] = {3,4,5,6,7}; // Entrada de pinos de configuração aqui bit 1 a 5
3 int dec_position = 0;
4
5 boolean a = 0;
6 void setup() {
7
8   Serial.begin(9600);
9
10  for(byte i = 0; i < 5; i = i + 1 ){ // Declaração dos pinos
11    pinMode(input_pin[i], INPUT);
12  }
13
14 }
15
16 void loop() {
17   Read_input(); // Leitura do sinal de Encoder
18   delay(100);
19
20 }
21 void Read_input() {
22   for(byte i = 0; i < 5; i = i + 1 ){
23     pin_state[i] = digitalRead(input_pin[i]);
24   }
25
26   // Decodificação do código Gray do disco
27   dec_position = pin_state[4];
28   for( int i = 3; i >= 0; i = i - 1){
29     dec_position = (dec_position << 1) | (pin_state[i] ^ (dec_position & 0x1));
30   }
31
32   Serial.print("RAW position: "); // o encoder é de 5 bits então contém 32 possíveis posições de 0 a 360 graus
33   Serial.println(dec_position, DEC);
34   Serial.print("Position in degree: ");
35   Serial.print(dec_position * 11.25); // 360 / (2^5) = 11,25 graus para cada posição das 32 posições possíveis
36   Serial.println("");
37 }
```



#### 4. Resultados e Discussão

O encoder absoluto foi montado, tendo o resultado final representado nas Figuras 8, 9 e 10.

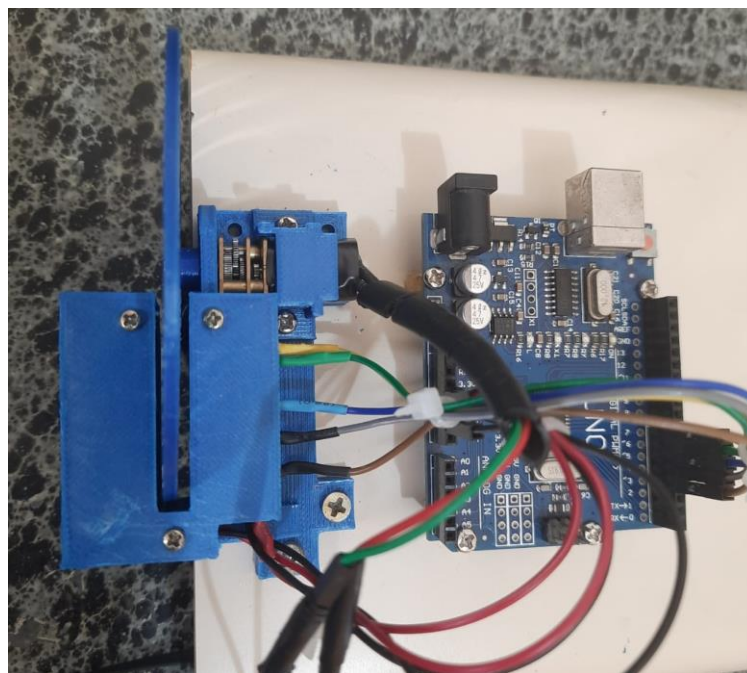


Figura 8 - Encoder montado (vista 1).

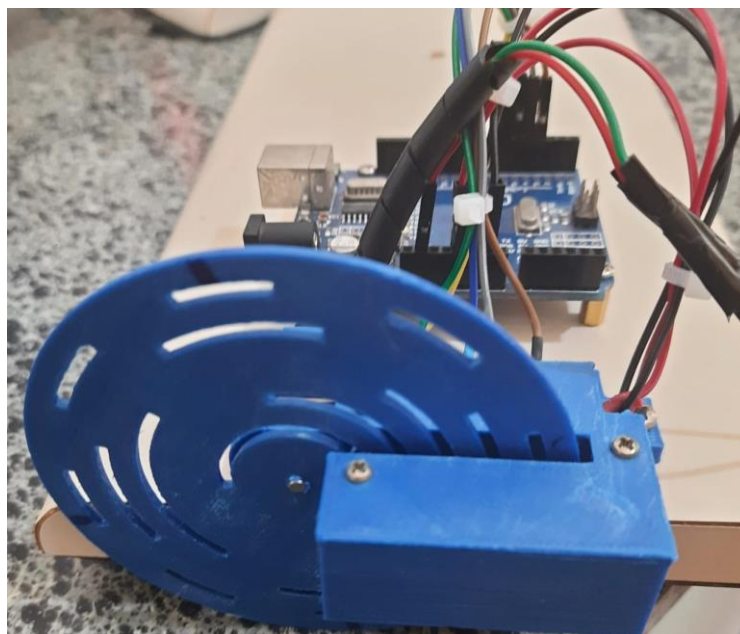


Figura 9 - Encoder montado (vista 2).

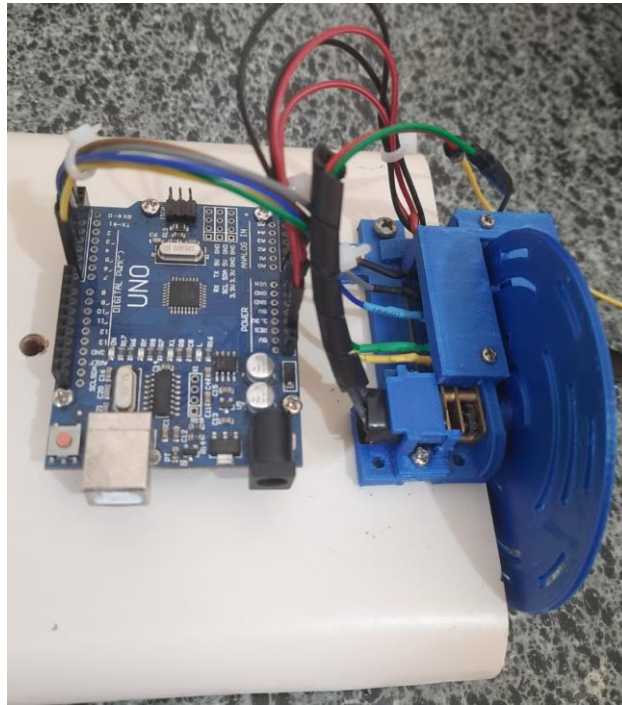


Figura 10 - Encoder montado (vista 3).

O encoder foi testado e atendeu muito bem minha expectativa, conseguindo obter todas as posições de forma precisa.

## 5. Conclusão

Com esse projeto de criação de um encoder absoluto concluo que foi essencial para exercitar conceitos fundamentais de eletrônica, robótica, lógicas de binário e código Gray e etc, afim de proporcionar uma experiência que agrega grande valor acadêmico, intelectual e profissional.

## 6. Referências Bibliográficas

- [1] [#03 Incremental type and absolute type | Basic Knowledge of Encoder | TUTORIALS | Asahi Kasei Microdevices \(AKM\).](#)
- [2] [EZAutomation Industry Article | High Speed Rotary Encoders.](#)
- [3] [Absolute Position Encoder with Non-Contact Magnetic Sensor \(rozum.com\)](#)
- [4] [Types of encoders. Sensing technology of rotary and linear encoder \(eltra-encoder.eu\)](#)

