**Dispenser de Balas Balistico - DBB©**

**Documentação**

**Autores**

Artur Assunção Costa

Daniel Curcio Lott Guimarães

Lucas Kelly Martins Lacerda

**Objetivo**

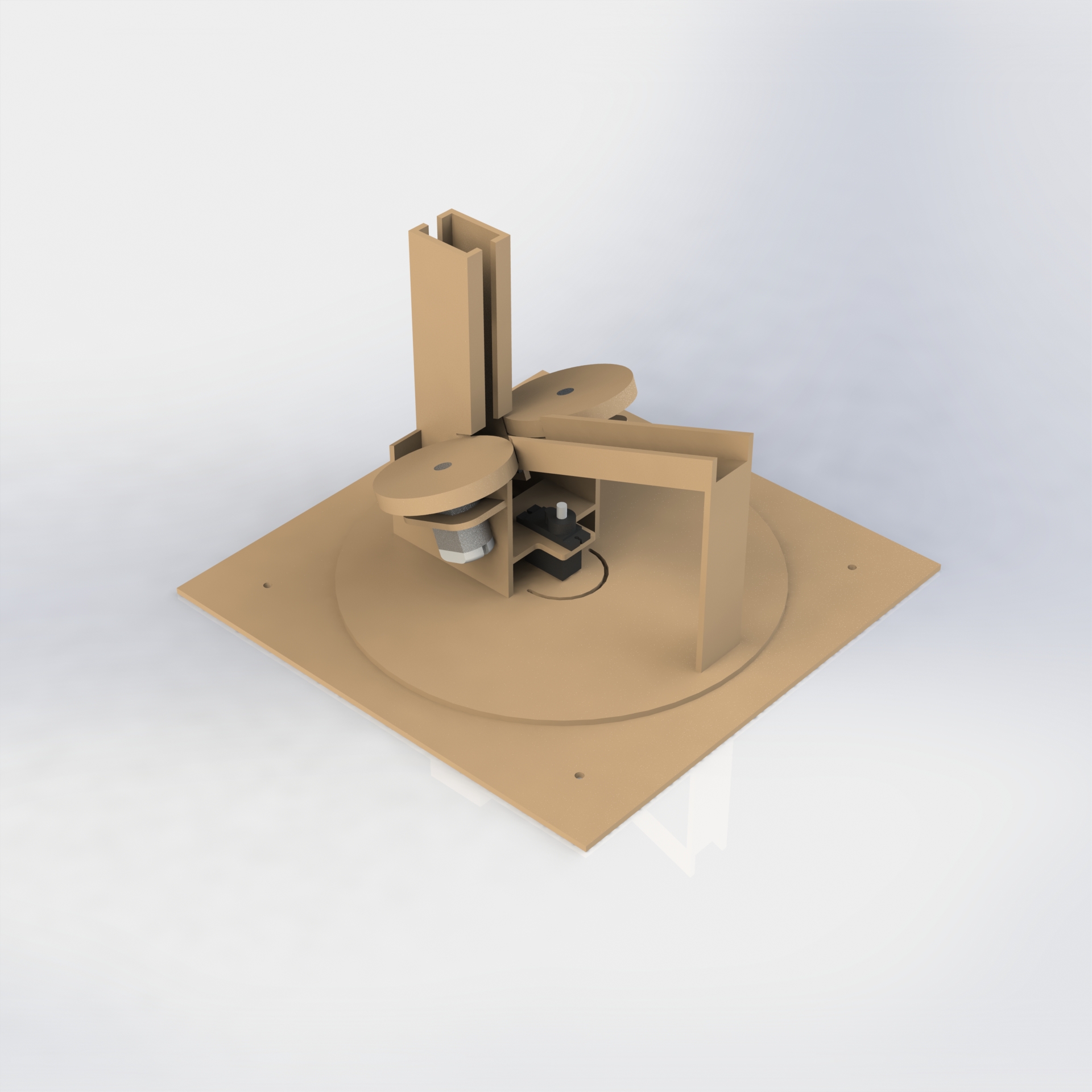
O equipamento a ser construído é capaz de detectar alvos em uma área varrida por 180°, de raio 30cm. Ao detectar qualquer objeto nessa área, o dispositivo dispara balas na direção do alvo, até que este saia de seu raio de alcance.

**Resumo:**

O DBB é um equipamento constituído por dois diferentes subsistemas: Detecção de alvos e arremesso de projéteis.

O subsistema de detecção de alvos funciona com um sensor ultrassônico acoplado a um servo que rotaciona uma base em 180°. O sensor avalia então se há algum objeto em um raio de 30cm. Caso algum objeto seja detectado por esse sistema, então o sistema de arremesso dos projéteis é acionado.

O subsistema de arremesso de projéteis consiste em um servo que movimenta uma bala em direção à dois motores CC que são acionados em sentidos opostos. Quando a bala passa pelos motores, ela é arremessada por uma rampa em direção ao objeto detectado pelo subsistema de detecção. Os motores CC são acionados através de um transistor NPN.

****

**Funcionamento**

Para cumprir com a missão, são necessários dois subsistemas: detecção e arremesso. Para melhor compreensão e eficiência, a construção do DBB© foi dividida em 4 partes: Detecção (Eletrônica), Arremesso (Eletrônica), Interface Gráfica (Eletrônica) e Mecânica (Material).

**Materiais Necessários**

* **1x Arduíno Uno**
* **Jumpers**
* **1x Transistor NPN bc639**
* **2x Resistores 180 ohms**
* **2x Motores DC**
* **1x Sensor Ultrassônico HC-SR04**
* **1x Protoboard**
* **1x Placa de Depron 5mm**
* **1x Diodo 1N4007**
* **Estiletes**
* **Cola quente**
* **Clipes de papel**

**Detecção**

O módulo de detecção consiste em um sensor ultrassônico e um servo para sua movimentação. Para o melhor funcionamento do sensor, utilizou-se nesse projeto a biblioteca <HCSR04.h>, já para o comando dos servos, utilizou-se a biblioteca <Servo.h>.

Antes de tudo, devemos escrever um programa que faça o servo movimentar-se em todo o seu alcance, e depois retornar à posição inicial. Isso é realizado com o seguinte código:

|  |
| --- |
| #include <Servo.h>  int angulo=-180;  void setup (){  Serial.begin(9600);  Servo1.attach(3);  }  void loop (){  Watch();  }  int Watch (){  int i=0;  for (i=angulo;i<180;i++){  Servo1.write(abs(i));  delay(20);  } |

Posteriormente, definimos a condição de detecção, que quando satisfeita, deverá dar início ao código de disparo. Definiu-se a condição de detecção em r <= 30cm.

|  |
| --- |
| #include <Servo.h>  #include <HCSR04.h>  UltraSonicDistanceSensor Torre (13,12);  Servo Servo1;  int angulo=-180;  void setup (){  Serial.begin(9600);  Servo1.attach(3);  }  void loop (){  Watch();  }  int Watch (){  int i=0;  float dist=100;  for (i=angulo;i<180;i++){  Servo1.write(abs(i));  delay(20);  if (i%2==0){  dist = Torre.measureDistanceCm();  }  if (dist<30.0){  while (Torre.measureDistanceCm()<30) {  dist = Torre.measureDistanceCm();  }  delay(100);  angulo=i;  return 1;  }  angulo=-180;  return 0;  } |

Repare que as portas 12 e 13 foram utilizadas para enviar e receber os dados do sensor, e a porta 3 foi utilizada para o comando do servo.

**Arremesso**

O arremesso das balas é feito em três comandos: um servo desloca a bala mais inferior em uma pilha de balas para uma região onde encontram-se dois motores rodando em sentidos opostos, de forma que ambos adicionam velocidade à bala, por atrito, atirando-a no sentido do alvo. Então, o servo recolhe seu braço para a posição inicial. Para que isso funcione, dentro da condição de detecção, inserimos linhas de código que, em ordem:

1. Acionam os motores
2. Movimentam o servo em direção aos motores
3. Recolhem o servo em direção à pilha, para um próximo lançamento

Enquanto a condição de detecção for satisfeita, esses comandos se repetem. Note que nessa condição, o servo de movimentação do detector se mantém parado. O código final ficará dessa forma:

|  |
| --- |
| #include <Servo.h>  #include <HCSR04.h>  UltraSonicDistanceSensor Torre (13,12);  Servo Servo1;  Servo Servo2;  int angulo=-180;  int LED1 = 9;  void setup (){  Serial.begin(9600);  Servo1.attach(3);  Servo2.attach(10);  Servo2.write(140);  pinMode(9,OUTPUT);  digitalWrite(9,LOW);  }  void loop (){  Watch();  }  int Watch (){  int i=0;  float dist=100;  for (i=angulo;i<180;i++){  Servo1.write(abs(i));  delay(20);  if (i%2==0){  dist = Torre.measureDistanceCm();  }  if (dist<30.0){  while (Torre.measureDistanceCm()<30) {  digitalWrite(9,HIGH);  delay(1000);  Servo2.write(80);  delay(100);  Servo2.write(135);  delay(1000);  dist = Torre.measureDistanceCm();  }  delay(100);  angulo=i;  digitalWrite(9, LOW);  return 1;  }  }  angulo = -180;  return 0;  } |

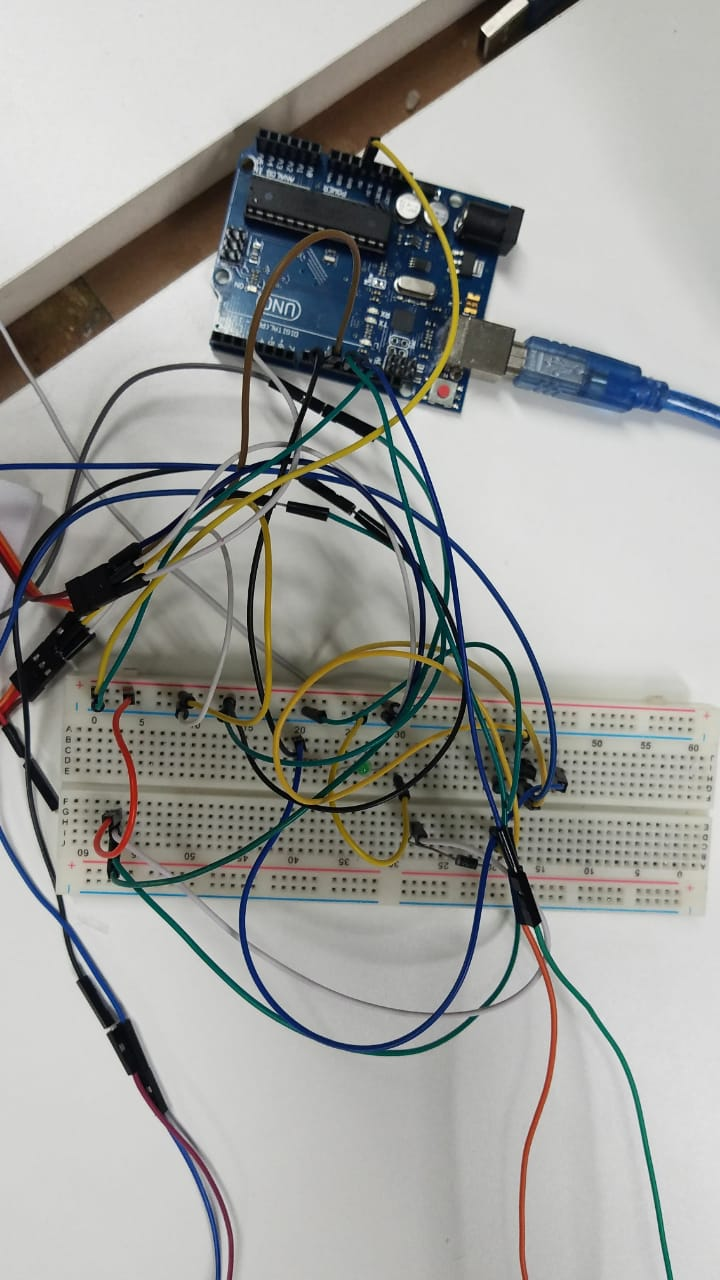
Para o acionamento dos motores, utilizou-se um transistor e um diodo de roda livre. Com essa montagem, é possível controlar o acionamento dos motores, e quando os mesmos são desligados, eles não têm uma desaceleração abrupta. Para o comando do transistor, utilizou-se a porta 9, e para o servo que desloca a munição, a porta 10. O diagrama esquemático da parte eletrônica encontra-se anexado no final do documento.

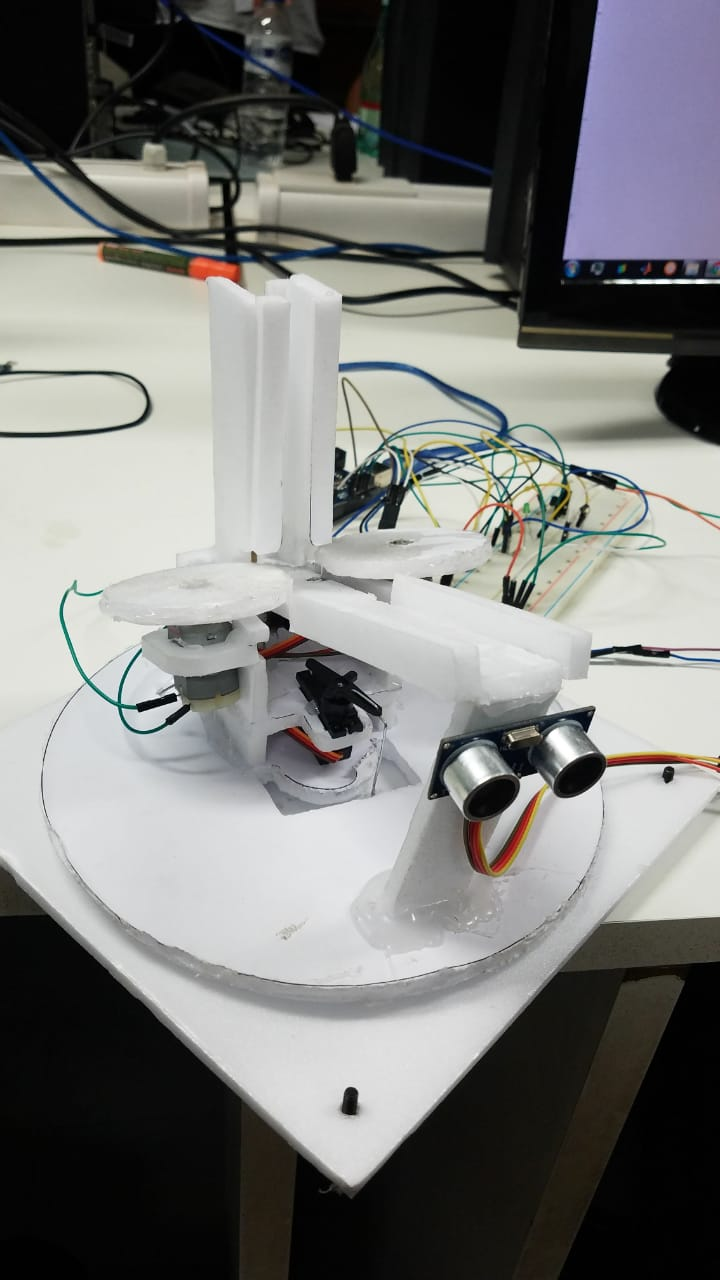
**Mecânica**

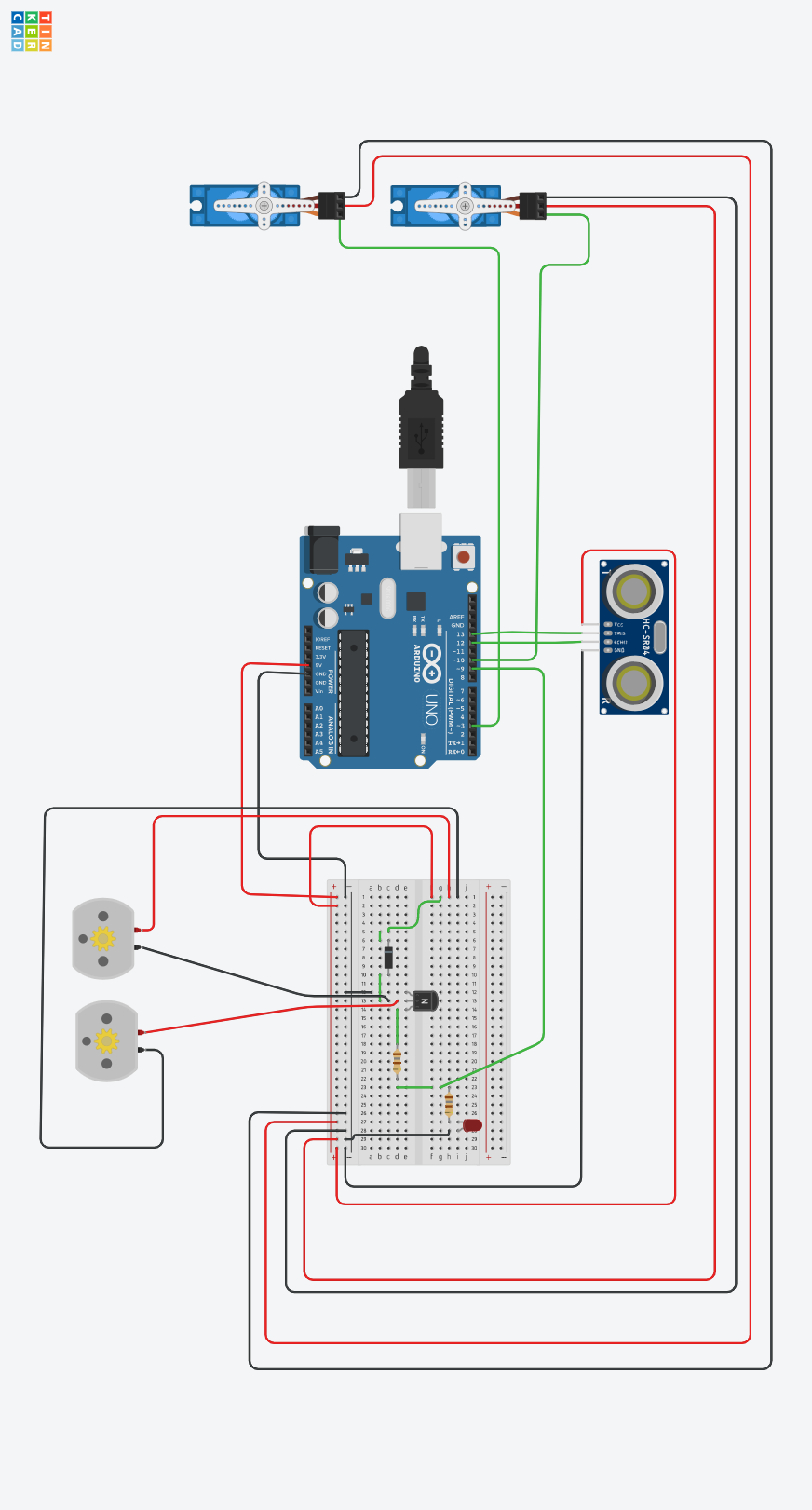
A parte mecânica consiste em uma base presa ao objeto Servo “Servo1” que é rotacionada com o torque causado por esse servo. Junto à essa base rotativa, estão acoplados o sensor ultrassônico e o dispenser de projéteis com a rampa e os motores.

Quando o sistema de detecção detecta algum alvo no raio especificado, os motores são acionados, e um servo movimenta o projétil em direção a eles. Foram presos aos motores cilindros responsáveis por aumentar a velocidade dos projéteis.

Para alimentar o disparador, construiu-se uma torre que comporta 8 balas empilhadas, e para o correto direcionamento dos disparos, elaborou-se uma rampa. As fotos da montagem final seguem em anexo.



****

****