

Vinícius Victor Giroto

**Laboratório de Sistemas Computacionais:
Comunicação Digital
Utilizando Arduino e comunicação Bluetooth
para jogar o jogo do dinossauro do Google**

São José dos Campos - Brasil

Agosto de 2021

Vinícius Victor Giroto

**Laboratório de Sistemas Computacionais:
Comunicação Digital**
**Utilizando Arduino e comunicação Bluetooth para jogar o
jogo do dinossauro do Google**

Relatório apresentado à Universidade Federal de São Paulo como parte dos requisitos para aprovação na disciplina de Laboratório de Sistemas Computacionais: Comunicação Digital.

Docente: Prof. Dr. Lauro Paulo da Silva Neto

Universidade Federal de São Paulo - UNIFESP

Instituto de Ciência e Tecnologia - Campus São José dos Campos

São José dos Campos - Brasil

Agosto de 2021

Resumo

As telecomunicações são partes essenciais das sociedades contemporâneas, estando presentes nos mais variados setores e sendo utilizada nas mais variadas atividades. Os meios de comunicação à distância foram primordiais para possibilitar a globalização do mundo e o estreitamento das relações humanas. Além disso, diversos projetos só são possíveis graças à essas tecnologias, como os aparelhos celulares e a internet, fundamentais no mundo atual. Assim, uma dessas tecnologias que vêm transformando o mundo é o Bluetooth, meio de comunicação sem fio que permite a transmissão de dados em curtos alcances e que pode ser utilizado livremente em tecnologias caseiras. Portanto, esse projeto busca utilizar o poder da tecnologia Bluetooth, aliado à ferramenta de hardware e software Arduino para automatizar o jogo do dinossauro do Google, famoso por aparecer durante falhas da internet. O trabalho consistiu na montagem de circuitos no Arduino que utilizam sensor de luminosidade e o Bluetooth para informar ao computador os momentos certos de se apertar uma tecla, o que é automatizado com o auxílio da linguagem de programação Python. Ao fim dos testes constatou-se o sucesso do projeto, tendo conseguido atingir mais de 500 pontos no jogo e cumprir o objetivo inicial do trabalho de automatização da jogabilidade.

Palavras-chaves: Bluetooth. Arduino. automatizar jogo. jogo dinossauro.

Lista de ilustrações

Figura 1 – Arduino e seus componenes auxiliares	10
Figura 2 – Jogo do Dinossauro	12
Figura 3 – Circuito para conexão do módulo Bluetooth no Arduino	14
Figura 4 – Módulo Bluetooth	14
Figura 5 – Circuito do Sensor de Luminosidade LDR	15
Figura 6 – Circuito do LED	15
Figura 7 – Foto do circuito completo	16
Figura 8 – Foto do sensor fixado no monitor	17
Figura 9 – Foto do jogo em execução com o LED ativado	21
Figura 10 – Foto do jogo em execução no momento da ação	22
Figura 11 – Foto do jogo no momento de um erro	23

Sumário

1	INTRODUÇÃO	5
2	OBJETIVOS	7
2.0.1	Objetivos gerais	7
2.0.2	Objetivos específicos	7
3	FUNDAMENTAÇÃO TEÓRICA	9
3.1	Bluetooth	9
3.2	Arduino	9
3.3	Python	11
3.4	Jogo do dinossauro do Google	11
4	DESENVOLVIMENTO	13
4.1	Materiais Utilizados	13
4.2	Circuito Arduino	13
4.2.1	Conexão Bluetooth	13
4.2.2	Sensor de Luminosidade LDR	15
4.2.3	Circuito do LED	15
4.3	Sistema Interligado	16
4.4	Código Arduino	17
4.5	Código Python	18
5	DISCUSSÃO DOS RESULTADOS	21
5.1	Automatizando o Jogo	21
5.2	Discussão dos Testes	23
	REFERÊNCIAS	25

1 Introdução

Ao longo da história foi necessário que o ser humano inventasse diferentes meios de se comunicar com outras pessoas. Passando por pinturas rupestres, envio de cartas, conversas por telefone, até chegar nas tecnologias mais atuais, como o envio de arquivos complexos instantaneamente, o homem sempre buscou aperfeiçoar os meios de comunicação para suprir a necessidade de contato que a sociedade tem.

Assim, o estudo das técnicas de envio de sinais elétricos é essencial para a existência do mundo como é hoje. E para isso existe a telecomunicação que, segundo (1), é:

(...) o ramo da engenharia elétrica que trata do projeto, da implantação e da manutenção dos sistemas de comunicações e têm por objetivo principal atender à necessidade do ser inteligente de se comunicar a distância.

Portanto, através das telecomunicações o mundo teve uma das suas maiores mudanças, a revolução da informação, que possibilitou a globalização do mundo e o estreitamento das relações humanas. Os meios de comunicação, principalmente sem fio, transformaram as rotinas humanas, facilitando a vida das pessoas e possibilitando novas dinâmicas nas atividades de lazer, trabalho e estudos.

Aliado aos meios de comunicação, a evolução dos computadores e dispositivos relacionados foram fundamentais para a revolução da informação, pois esses foram responsáveis por fornecer o *background* necessário para a existência das telecomunicações. Os dispositivos de *hardware* oferecem recursos que permitem fazer o tratamento dos sinais elétricos passados de uma fonte para outra, facilitando a manutenção desses dados e possibilitando que os desenvolvedores utilizem as melhores técnicas para obter boa eficiência nas comunicações.

Em vista disso, esse relatório apresenta a interligação do dispositivo de *hardware* Arduino com a tecnologia de comunicação Bluetooth para automatizar um jogo. Para isso, as seções desse relatório se dividem da seguinte forma: a seção **Objetivos** exibe o objetivo desse projeto, especificando as metas visadas; a seção **Fundamentação Teórica** apresenta os principais conceitos sobre Arduino, a tecnologia Bluetooth, a linguagem de programação Python e o jogo utilizado para o projeto; a seção **Desenvolvimento** apresenta as principais ferramentas usadas para o desenvolvimento do projeto e exibe todos os processos de implementação do trabalho; por fim, a seção **Discussão dos Resultados** apresenta os principais resultados obtidos do projeto, apresentando o sistema funcionando e os resultados dos testes.

2 Objetivos

2.0.1 Objetivos gerais

Nesse trabalho objetiva-se desenvolver um sistema utilizando Arduino e comunicação Bluetooth para enviar um sinal para o computador e através do Python automatizar o jogo do dinossauro do Google.

2.0.2 Objetivos específicos

Os objetivos específicos do trabalho são:

- Montar circuito do módulo bluetooth no arduino;
- Montar circuito do sensor de luminosidade no arduino e prende-lo ao monitor;
- Montar circuito do LED no arduino;
- Programar o arduino para ler o sensor e enviar saídas para o LED e por bluetooth para o computador;
- Conectar o computador por bluetooth ao arduino;
- Programar aplicação em python para ler o sinal bluetooth e acionar comando no jogo;
- Fazer testes de adequação e aperfeiçoamento da jogabilidade;

3 Fundamentação Teórica

3.1 Bluetooth

O Bluetooth, também chamado de IEEE 812.15.1, é uma tecnologia criada para prover comunicação de rádio de curto alcance e baixo consumo de energia. Ele foi planejado e é mantido até hoje pelo *Bluetooth Special Interest Group* (SIG), que inicialmente era formado pelas empresas Ericsson, Intel, Nokia, Toshiba and IBM, mas que atualmente conta com mais de 36,000 empresas focadas em aperfeiçoar a tecnologia e espalha-la para mais dispositivos (2) (3).

Sendo uma das tecnologias de comunicação sem fio mais famosas, o Bluetooth abrange uma variedade de aplicações, como transmissão de áudio em tempo real, transferência de arquivos entre dispositivos, serviços de localização, manutenção de redes de dispositivos, entre outras atividades, sendo utilizada em diversos tipos de dispositivos.

O Bluetooth não requer licença para uso, e se situa na banda de frequência ISM (*industrial, scientific, and medical*), atuando na faixa de frequência entre 2,40 GHz à 2,48 GHz. Ele usa a técnica de transmissão de sinais de rádio chamada FHSS (*Frequency hopping Spread Spectrum*) que consiste na mudança constante da frequência da portadora através de vários canais de frequência, podendo passar por qualquer um dos 79 canais do Bluetooth, cada um com intervalos de 1 MHz, sendo capaz de realizar até 1600 saltos por segundo ao utilizar a técnica de *adaptive frequency-hopping* (AFH) (4) (5).

3.2 Arduino

O Arduino é uma plataforma eletrônica usada para implementar, de forma fácil, aplicações de hardware e software. Tanto a placa eletrônico quanto o ambiente de desenvolvimento são de código aberto e voltados para experimentações e melhorias da comunidade (6).

Por ser de fácil uso e relativamente barato, o Arduino é utilizado por milhares de pessoas em todo o mundo. Sejam professores, estudantes, curiosos ou desenvolvedores, qualquer pessoa pode trabalhar essa plataforma e desenvolver inúmeros projetos, que abrangem diversas áreas e atividades, como automações, música, jogos, saúde, educação, entre muitas outras.

Todas essas aplicações são possíveis graças à grande variedade de sensores e dispositivos auxiliares que são possíveis de se utilizar junto ao Arduino. Sensores de temperatura, contato, luminosidade e diversos outros fenômenos contribuem para abrigar os projetos

3.3 Python

O Python é uma linguagem de programação muito poderosa, rápida, versátil e fácil de se utilizar. Ela possui código aberto e é livre para uso e distribuição, ela pode rodar em múltiplas plataformas e ser aplicada em diversos projetos, que variam em muitas áreas, dependendo apenas do desejo do programador (7).

O Python foi criado em 1989 pelo programador Guido Van Rossum, que desejava desenvolver uma linguagem com sintaxe simples e que suprisse as necessidades que ele via em outra linguagem. Assim, hoje o Python é uma das linguagens mais utilizadas em todo o mundo, sendo aplicada em projetos de ciência de dados, aplicações web, desenvolvimento de jogos, automações, entre outros que abrangem diversos setores (8).

Essa grande versatilidade do Python se deve, principalmente, à possibilidade de implementação e utilização de bibliotecas de códigos escritas por outras pessoas, além do código original da linguagem, permitindo que qualquer programador possa desenvolver códigos complexos que cumpram algum objetivo antes não executado pelo Python, e possa, então, disponibilizar esses códigos para qualquer interessado utilizar e, se preciso, aperfeiçoar, oferecendo uma gama de possibilidades aos usuários dessa linguagem.

3.4 Jogo do dinossauro do Google

O jogo do dinossauro é um pequeno jogo disponibilizado pelo navegador Google Chrome para o usuário se entreter nos momentos em que estivesse sem sua internet funcionando. Assim, ele é um jogo simples, que costuma aparecer no centro do navegador quando não há a internet, e que se inicia quando o jogador aperta a tecla de espaço. A tela de falta de internet com o jogo se iniciando é apresentada na [Figura 2](#).

Figura 2 – Jogo do Dinossauro



Fonte: Tectudo ([9](#)).

O objetivo do jogo é extremamente simples, apenas alcançar o maior número de pontos, obtidos automaticamente ao longo do tempo. Porém, se o jogador atingir uma árvore ou pássaro, ele morre, e então tem que recomeçar o jogo desde o início. Ou seja, para obter a maior pontuação, é necessário desviar dos obstáculos pulando ou abaixando quando eles aparecem, se atentando ao aumento de velocidade gradual existente no jogo, o que aumenta gradativamente a dificuldade de ficar vivo.

4 Desenvolvimento

4.1 Materiais Utilizados

Para o desenvolvimento de todo o projeto, foram utilizando os seguintes materiais e tecnologias:

- Placa Arduino UNO R3;
- Componentes do Arduino:
 - Placa protoboard;
 - Sensor de luminosidade LDR;
 - Módulo Bluetooth;
 - 5 resistores de 220 Ω ;
 - 1 LED verde;
 - 24 jumpers;
- Computador com tecnologia Bluetooth;
- IDE Arduino instalada;
- Python e biblioteca Pyserial instalados;
- Site "chrome://dino/" para executar o jogo.

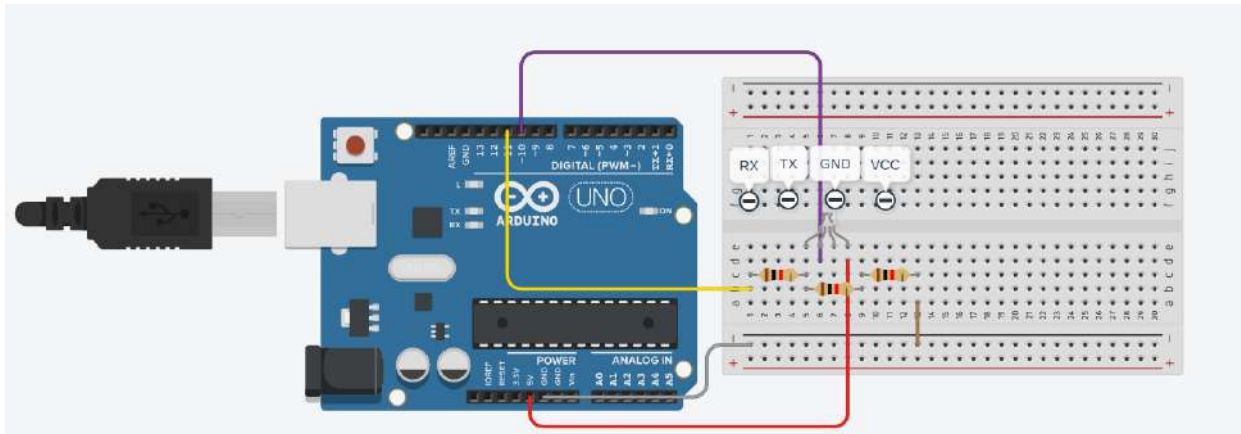
4.2 Circuito Arduino

O circuito do Arduino foi montado na protoboard interligando os jumpers às entradas do Arduino. Para mais fácil entendimento, foi montado o circuito de um componente por vez, obtendo no final todos os componentes ligados ao mesmo tempo. Assim, as subseções a seguir apresentam o processo de montagem dos circuitos utilizados.

4.2.1 Conexão Bluetooth

A [Figura 3](#) apresenta o primeiro circuito montado, referente ao circuito do módulo Bluetooth.

Figura 3 – Circuito para conexão do módulo Bluetooth no Arduino



Fonte: O Autor.

No site utilizado para montagem dos circuitos de demonstração ¹ não existe o módulo Bluetooth do Arduino, por isso foi utilizado um componente genérico para representá-lo. Assim, a Figura 4 apresenta uma foto do módulo Bluetooth com suas entradas escritas.

Figura 4 – Módulo Bluetooth



Fonte: O Autor.

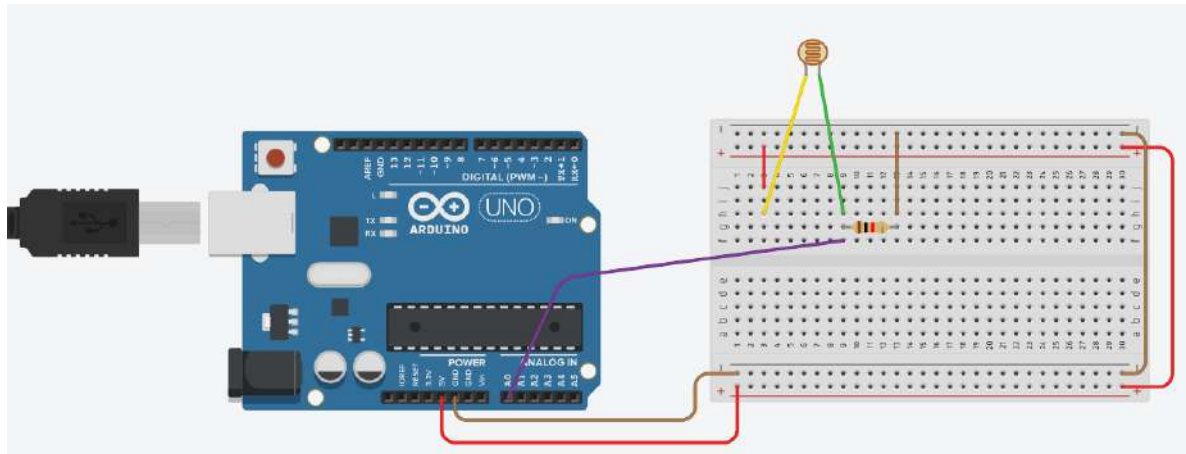
Portanto, observando as Figuras 3 e 4 podemos visualizar a montagem do circuito do módulo Bluetooth. Inicialmente, foram ligadas as entradas GND e VCC do módulo às entradas GND e VCC do Arduino, respectivamente. Em seguida, a porta TX do módulo, que é a porta transmissora de informação, foi ligada na porta digital 10 do Arduino. Por fim, para conexão da porta receptora (RX) do módulo, precisamos montar um circuito divisor de tensão para converter os 5 V enviados pelo Arduino para 3,3 V, que impede que o módulo Bluetooth seja danificado.

¹ <<https://www.tinkercad.com/>>

4.2.2 Sensor de Luminosidade LDR

A Figura 5 apresenta o circuito montado para utilização do sensor de luminosidade LDR.

Figura 5 – Circuito do Sensor de Luminosidade LDR



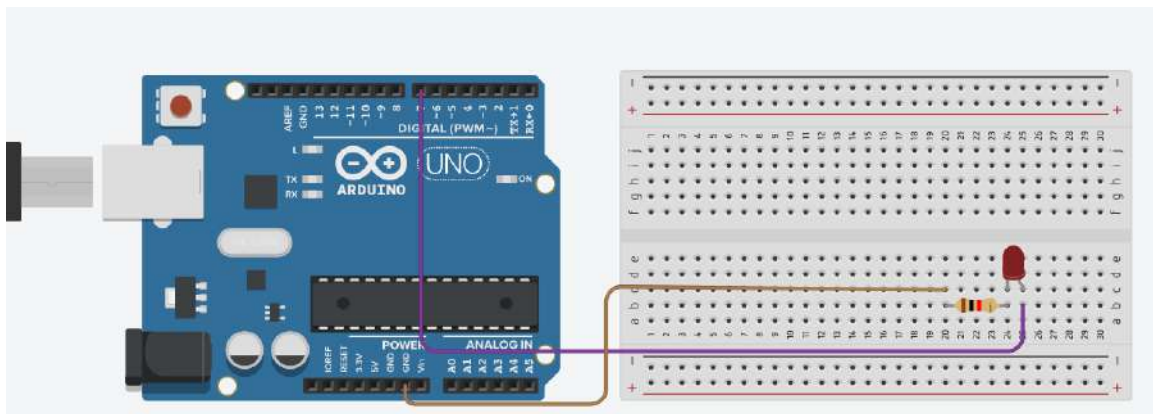
Fonte: O Autor.

Como pode se observar, para utilizar o sensor LDR são necessários poucos componentes, sendo conectado à uma perna do sensor a fonte de 5 V provida pelo Arduino, e à outra perna um circuito de *pull-down* com a entrada analógica A0 ligada ao GND através de um resistor.

4.2.3 Circuito do LED

Por fim, o último circuito montado foi o do LED. Como podemos ver na Figura 6, o circuito consiste da ânodo do LED ligada ao pino 7 do Arduino e da catodo ligada ao GND através de um resistor.

Figura 6 – Circuito do LED

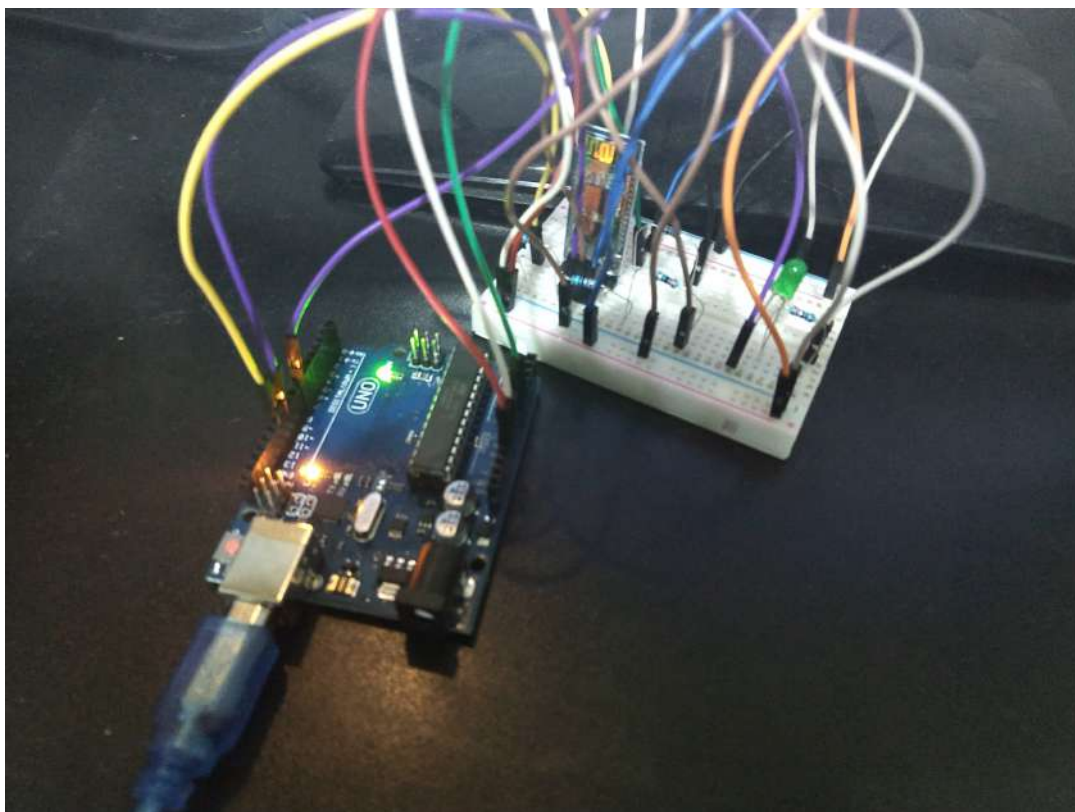


Fonte: O Autor.

4.3 Sistema Interligado

Após a montagem dos três circuitos anteriores, a parte de montagem do projeto foi concluída, possuindo os três circuitos montados e funcionando ao mesmo tempo, inclusive influenciando uns nos outros. A [Figura 7](#) apresenta uma foto dos circuitos montados na protoboard e conectados ao Arduino, com todos os fios posicionados da maneira que foram mostrados anteriormente.

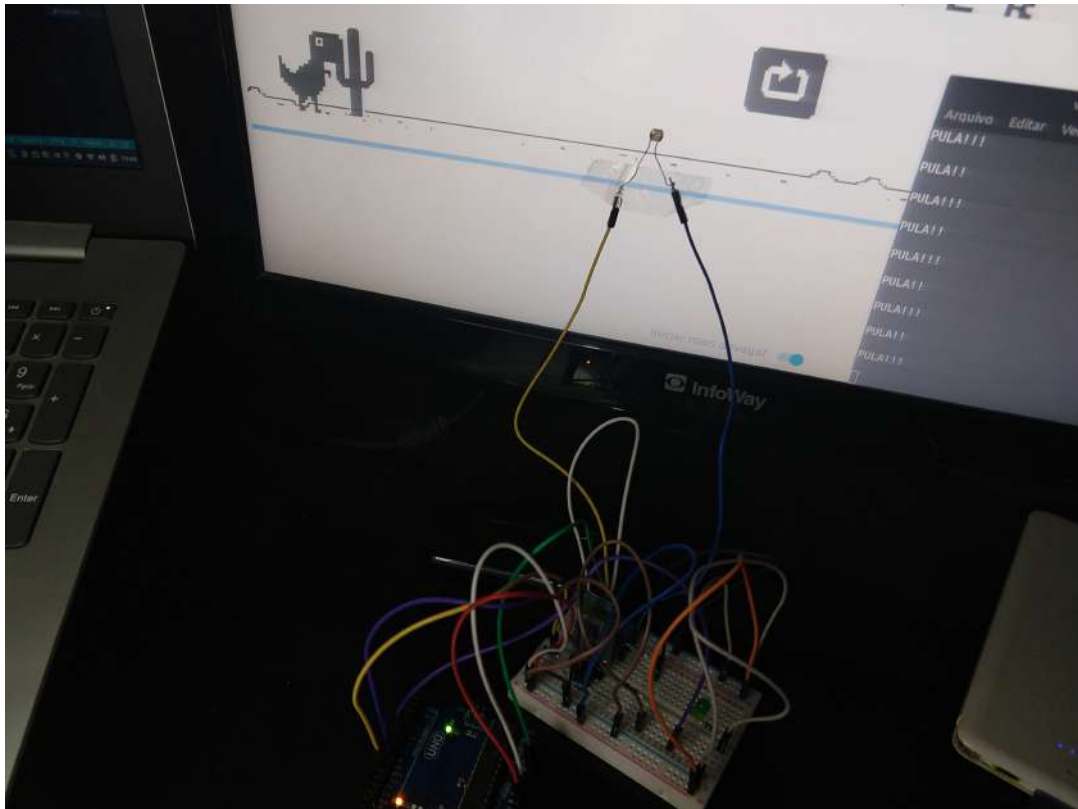
Figura 7 – Foto do circuito completo



Fonte: O Autor.

Para finalização das configurações do hardware, foi necessário prender o sensor LDR no monitor do computador, para que fosse possível a utilização contínua do sistema com o sensor na mesma posição. Além disso, foi necessário aumentar o brilho da tela do monitor ao máximo, para facilitar para o sensor distinguir os objetos claros e escuros na tela. Portanto, a [Figura 8](#) apresenta uma foto do sensor LDR colado ao monitor com uma fita adesiva. Note, também, a posição que foi fixado o sensor, em um local em que pudesse identificar tanto as árvores baixas quanto as árvores altas, mas que não consegue identificar os pássaros que voam alto, sendo esse um ponto de melhoria do projeto.

Figura 8 – Foto do sensor fixado no monitor



Fonte: O Autor.

4.4 Código Arduino

O código implementado na IDE do Arduino e executado no hardware dessa plataforma é apresentado a seguir.

Código executado no Arduino

```
1 #include <SoftwareSerial.h>
2 SoftwareSerial Bluetooth(10,11); // tx, rx (bluetooth)
3
4 int sensor = 0;
5 int state = 0;
6 int ledPin = 7;
7 int cont = 0;
8 int ledState = 0;
9
10 void setup() {
11     pinMode(ledPin, OUTPUT);
12     Bluetooth.begin(9600);
13 }
14
15 void loop() {
16
17     state = analogRead(sensor);
18
19     if (state < 53) {
```

```
20   Bluetooth.write("1\r\n");
21   digitalWrite(ledPin, HIGH);
22   ledState = 1;
23 }
24 else {
25   if ( ledState == 1){
26     if (cont > 100){
27       digitalWrite(ledPin, LOW);
28       cont = -1;
29       ledState = 0;
30     }
31     cont++;
32   }
33
34 }
35 }
```

Fonte: O Autor.

Como pode-se observar, inicialmente é importada para o código a biblioteca *SoftwareSerial*, utilizada para a comunicação Bluetooth. Em seguida, é criada uma instância serial para a conexão Bluetooth, especificando as portas utilizadas no Arduino. Após isso, são declaradas variáveis e no *setup* é inicializada a porta serial do Bluetooth e o pino do LED.

Na função do *loop*, que roda à todo momento, primeiramente é feita a leitura do sensor LDR, recebendo o valor de luminosidade lida e armazenando na variável *state*. Em seguida, faz-se a verificação do valor da variável *state*, se ela for menor que 53, para nossos testes, significa que o sensor encontrou um obstáculo no caminho, logo, o código envia um sinal por Bluetooth ao computador, liga o LED e define a variável *ledState* como 1.

Caso o valor de *state* seja igual ou maior que 53, o código entra no bloco *else*, e então, se a variável *ledState* for 1, é realizado uma contagem de 100 ciclos do *loop* para então desligar o LED. Essa parte do *else* existe para manter o LED ligado por mais tempo, e não apenas no momento em que o LDR encontra um obstáculo.

4.5 Código Python

O código Python para leitura do sinal Bluetooth e resposta com comando no jogo é apresentado a seguir.

Código executado no Python

```
1 import time
2 import serial
3 import pyautogui
4
5 # Iniciando conexao serial bluetooth
6 comport = serial.Serial('/dev/rfcomm0', 9600)
7
```

```
8 time.sleep(1.8) # Entre 1.5s a 2s
9 cont = 0
10
11 while ( True ):
12
13     VALUE_SERIAL=comport.readline()
14
15     x = str(VALUE_SERIAL)[2:3]
16
17     if x == '1':
18         pyautogui.press('space')
19         cont += 1
20         if cont % 2 == 0:
21             print("PULA!!\n")
22         else:
23             print("PULA!!!\n")
24
25
26     comport.reset_input_buffer()
```

Fonte: O Autor.

Inicialmente são importadas as bibliotecas `time`, para pausar o código, `serial`, para fazer leitura do serial do computador, e `pyautogui`, para automatizar o pressionamento de teclas. Após isso, é instanciada a entrada serial do computador especificando o diretório do Bluetooth e definido o *baud rate* como 9600. Então é feita a espera de 1,8 segundos até iniciar o ciclo *while*.

O ciclo *while* é rodado indefinidamente, sendo parado apenas pela interrupção do código. Assim, dentro do laço primeiro é feita a leitura do *buffer* do Bluetooth, em seguida, a variável *x* recebe apenas o terceiro caractere passado, que, caso tenha sido enviado pelo Arduino, é o número 1. Logo, há a condição *if* que verifica se *x* possui o valor 1, caso possua, o `pyautogui` realiza o pressionamento da tecla espaço do computador, responsável pelo pulo no jogo. Além disso, é feito uma adição em um contador e a verificação se ele é par ou ímpar, apenas para imprimir uma string diferente na tela.

Por fim, ainda dentro do laço *while*, há o comando *reset input buffer* que limpa o *buffer* do Bluetooth e garante que não haverá nenhum bug devido ao Arduino ter mandado diversos sinais em sequência, atestando que quando encontrado algum obstáculo no jogo, o dinossauro só receberá um comando de pulo.

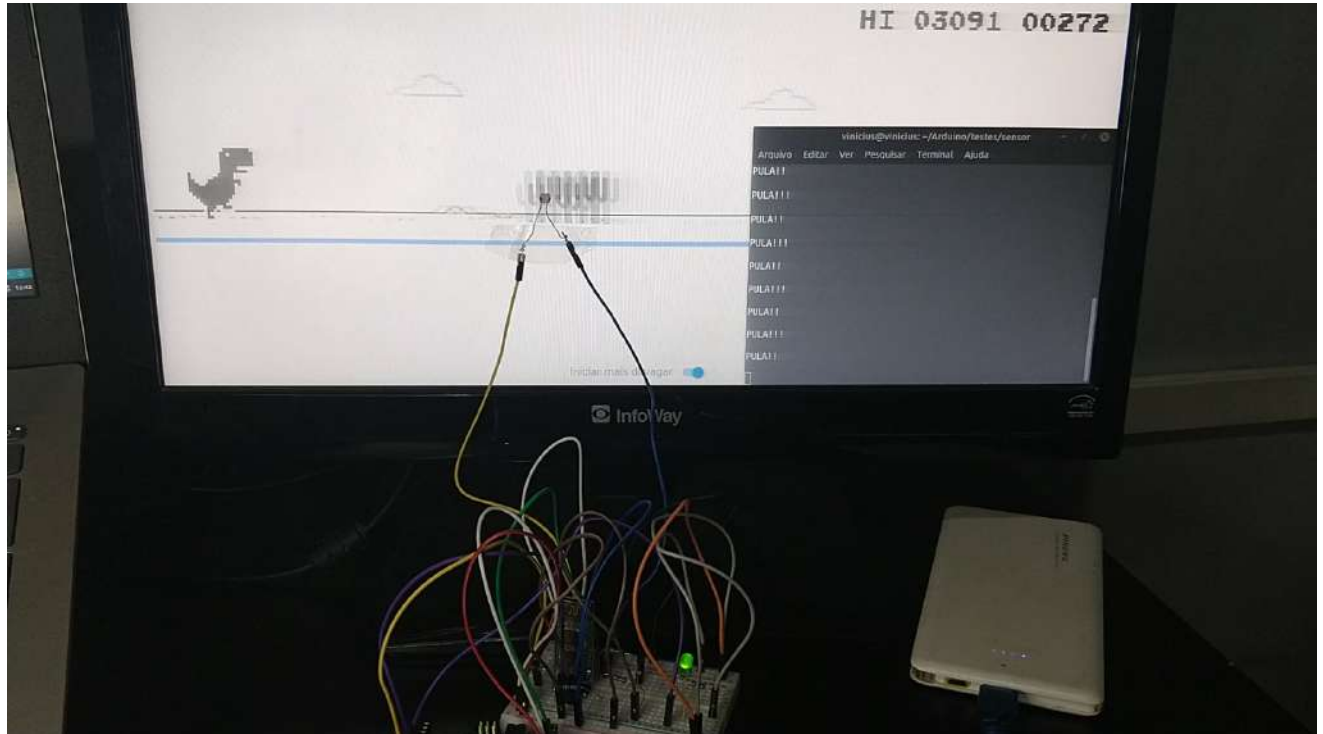
5 Discussão dos Resultados

5.1 Automatizando o Jogo

Após todos os circuitos terem sido implementados e integrados, o código do Arduino ter sido executado na plataforma e o código Python estar pronto para execução, os testes começaram a serem realizados.

Primeiramente foram feitos testes para ajustar a distância do sensor LDR para a posição do dinossauro no jogo, para obter o melhor tempo de reação possível. Durante os testes, notou-se que desde a detecção do obstáculo pelo sensor até o pressionamento da tecla pelo Python há um atraso de cerca de 0,35 segundos, por isso, o sensor foi colocado mais à frente para garantir um tempo de reação adequado. Na [Figura 9](#) podemos notar essa distância escolhida, e que o LED está aceso, significando que o sensor identificou um obstáculo, porém, ainda não foi dado o comando de pulo para o jogo.

Figura 9 – Foto do jogo em execução com o LED ativado



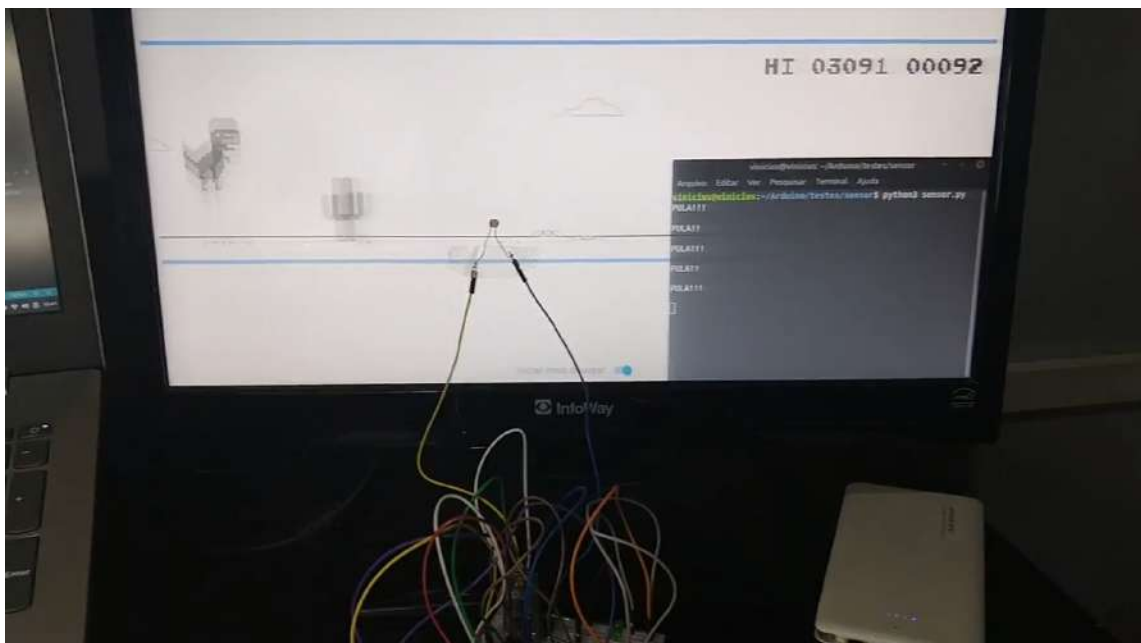
Fonte: O Autor.

Na [Figura 9](#) pode-se perceber também o algoritmo python sendo executado no terminal do linux na direita da tela. Como esperado, após realizar um pulo o código

imprime na tela uma string de indicação da ação e, dado que já foi passado um tempo de jogo, notável pela pontuação atual de 272, foram feito diversos *prints* na tela.

Já na [Figura 10](#) pode-se observar o momento do pulo do dinossauro, que aconteceu algum tempo depois do sensor ter identificado o obstáculo. Como é possível ver, o dinossauro consegue pular no tempo certo para evitar a colisão com a árvore, atestando que o sensor está localizado na posição ideal, pelo menos para o início do jogo. Observa-se, também, que o código Python acabou de ser executado, possuindo, no momento da foto, pouco impressões na tela.

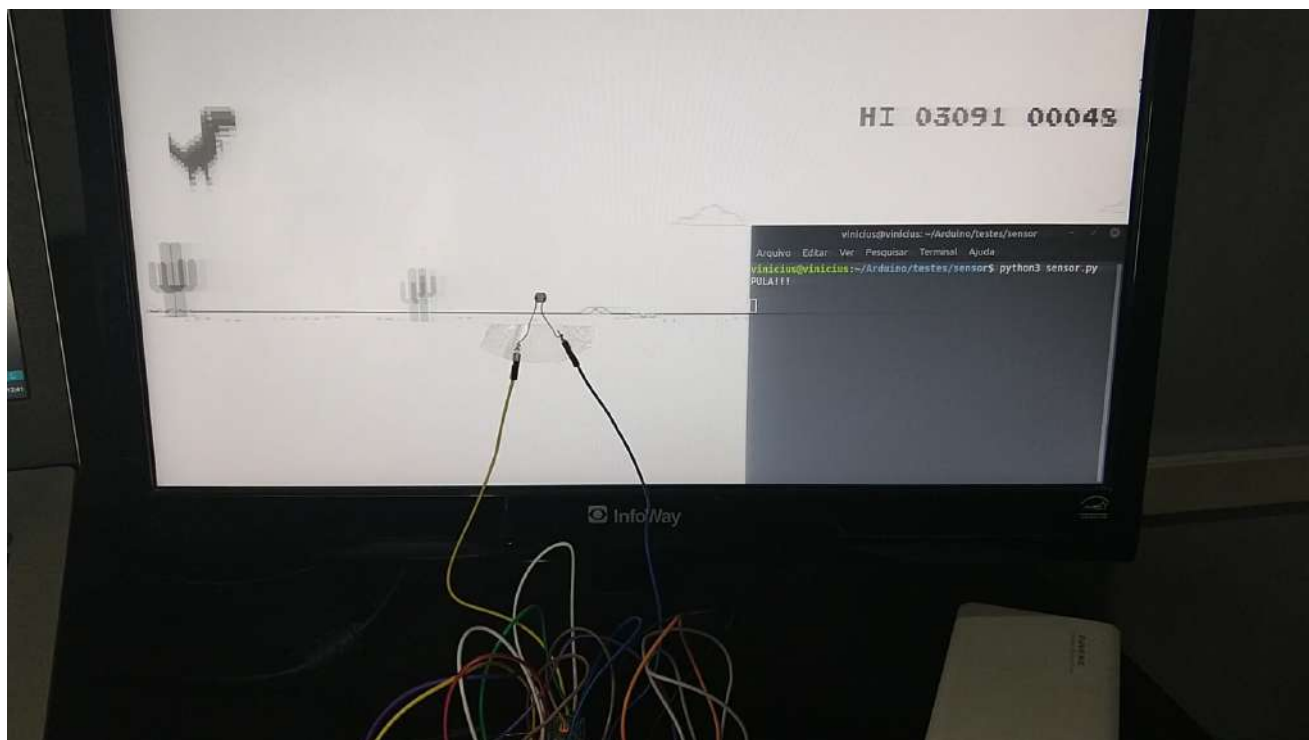
Figura 10 – Foto do jogo em execução no momento da ação



Fonte: O Autor.

Por fim, a [Figura 11](#) mostra uma situação em que o projeto não consegue desviar de um obstáculo, devido à ele não ter inteligência para perceber que há outro obstáculo logo em seguida, enquanto ainda está na pulo. Essa é uma situação pouco comum para o início do jogo, mas, após um certo tempo, passa a ser mais comum esse tipo de obstáculo, e então o projeto apresenta dificuldades para desviar.

Figura 11 – Foto do jogo no momento de um erro



Fonte: O Autor.

Há outras situações de erros que puderam ser notadas durante os testes, como casos em que o sensor não consegue identificar um obstáculo, talvez por ele passar rápido demais, e então o dinossauro não desvia. Ou, como mencionado anteriormente, o projeto tende a perder quando aparece o obstáculo de pássaro, que está na parte de cima da tela e, por isso, o sensor não consegue identificar.

5.2 Discussão dos Testes

Após inúmeros testes, o melhor resultado obtido foi de 570 pontos, o que, se comparado à um humano experiente jogando, é uma baixa pontuação, mas para uma automação é um ótimo resultado, considerando que foi preciso desviar de algumas dezenas de obstáculos. Assim, entende-se que o objetivo do projeto foi alcançado, ao conseguir automatizar o jogo e, ainda, obter uma alta pontuação.

No entanto, considera-se que ainda existem pontos de melhoria no trabalho, como a possibilidade de identificar pássaros e alguma adequação para tratar o *atraso* de resposta do jogo, podendo fixar o sensor mais à frente em relação ao dinossauro e, então, controlar por código o tempo de espera para o pulo. Dessa forma, seria possível também fazer a adequação para os pulos acontecerem mais rápidos conforme o jogo avança, para tratar a aceleração que ocorre gradualmente na velocidade.

Referências

- 1 MEDEIROS, J. C. de O. *Princípios de telecomunicações: teoria e prática*. 2th edition. ed. São Paulo: Érica, 2007. Citado na página 5.
- 2 BLUETOOTH SIG, INC. *Bluetooth*. Disponível em: <<https://www.bluetooth.com/>>. Acesso em: 13 ago. 2021. Citado na página 9.
- 3 SIQUEIRA, T. S. de. Bluetooth – características, protocolos e funcionamento. Disponível em: <<https://ic.unicamp.br/~ducatte/mo401/1s2006/T2/057642-T.pdf>>. Acesso em: 13 ago. 2021. Citado na página 9.
- 4 NETO, L. P. da S. Espalhamento espectral e comunicações por bluetooth. Unifesp/ICT. Acesso em: 13 ago. 2021. Citado na página 9.
- 5 WIKIMEDIA FOUNDATION, INC. *Bluetooth*. Disponível em: <<https://en.wikipedia.org/wiki/Bluetooth>>. Acesso em: 13 ago. 2021. Citado na página 9.
- 6 ARDUINO. *What is Arduino?* 2021. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 14 ago. 2021. Citado na página 9.
- 7 PYTHON SOFTWARE FOUNDATION. *Python - About*. Disponível em: <<https://www.python.org/about/>>. Acesso em: 14 ago. 2021. Citado na página 11.
- 8 GRUPO ALURA. *Python: A origem do nome*. Disponível em: <<https://www.alura.com.br/artigos/python-origem-do-nome>>. Acesso em: 14 ago. 2021. Citado na página 11.
- 9 GLOBO COMUNICAÇÃO E PARTICIPAÇÕES S.A. *Como jogar o Jogo do Dinossauro do Google online no navegador Chrome*. Disponível em: <<https://www.techtudo.com.br/dicas-e-tutoriais/2019/11/como-jogar-o-jogo-do-dinossauro-do-google-online-no-navegador-chrome.ghtml>>. Acesso em: 14 ago. 2021. Citado na página 12.