

# BtsA

## (Botões Acessíveis)

### Identificação

Nome: Daniel Santos de Sena

Disciplina/Turma: Fundamentos de Sistemas Embarcados - Turma 3

email acadêmico: [djdanielsan@academico.ufs.br](mailto:djdanielsan@academico.ufs.br)

# Objetivo/Motivação

- Projetar um firmware que juntamente com hardware barato, possamos levar periféricos e tecnologia para pessoas com a movimentação reduzida
- Os dispositivos de acessibilidade atuais são caros e muitas vezes inacessível
- Objetivo principal é criar uma alternativa barata em relação aos produtos existentes, criando produtos acessíveis para comunidades que não possuam recursos financeiros suficientes para comprar os produtos existentes no mercado de tecnologia assistiva.

# Fundamentação

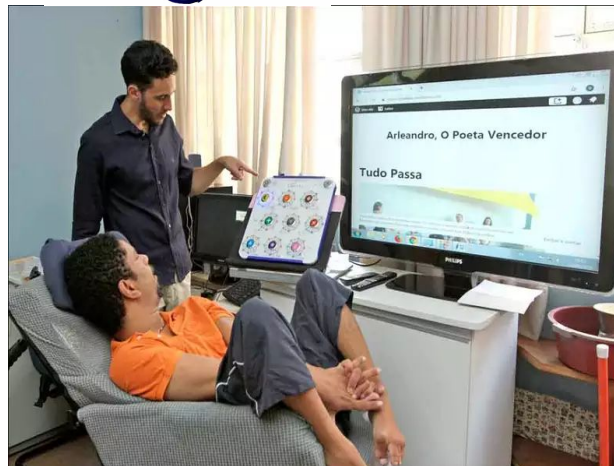
## Xbox Adaptive Controller

- Controles de Xbox produzidos pela própria Microsoft que podem receber periféricos personalizados.



## Teclado Inteligente TiX

- Base com 11 botões, na qual é possível utilizar mouse e teclado utilizando combinações entre estes botões.



## Botoeira Usb Cabine Fotográfica



# Fundamentação

## Preços

Xbox Adaptive Controller



## Xbox Adaptive Controller

Jogue do seu jeito.

R\$ 999\*

[ENCONTRE UM VAREJISTA >](#)

Teclado Inteligente TiX

A blue and white multifunctional keyboard with a grid of colorful buttons. A legend on the left lists functions for each color: amarelo (ampliar), laranja (antecar), verde (antecar), azul (antecar), vermelho (antecar), roxo (antecar), rosa (antecar), and cinza (antecar). A small blue cube is shown next to the keyboard.

Qual defici

R\$ 195/mês

PLANO AUTONOMIA

Use o Teclado Inteligente Multifuncional TiX para escrever no computador, controlar o mouse, navegar na internet, jogar jogos no computador, mesmo sem coordenação motora fina.

CLIQUE PARA MAIS DETALHES

# Funcionamento e Aplicação

- Poderá ser usado por pessoas que possuam mobilidade reduzida, possuindo o movimento voluntário das pernas ou dos braços/mãos, podendo ser personalizado com algum suporte para ser utilizado com movimentos voluntário da cabeça para direita ou esquerda.



## Simular as Teclas

- LEFT
- RIGHT
- UP
- DOWN



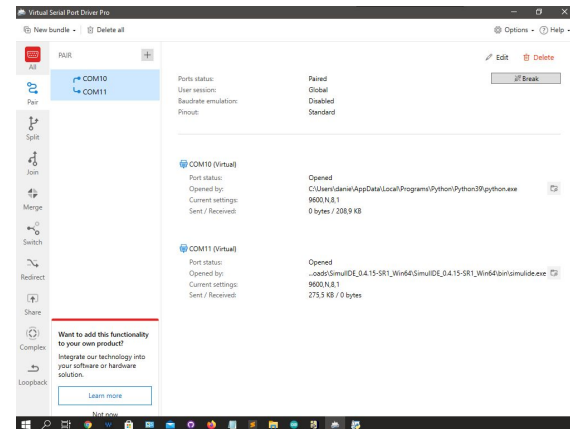
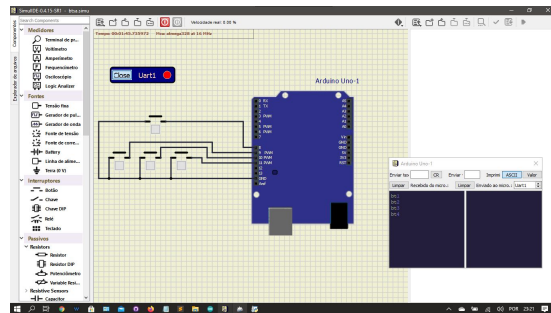
# Materiais e Ferramentas utilizados

- Arduino IDE  
Usado para criar o código do firmware que será executado pelo Arduino
- SimulIDE  
Utilizado para simular os hardwares, como o arduíno uno e os botões
- Python e as Bibliotecas
  - PySerial - Com o pyserial é possível receber os códigos que são enviados por algum dispositivo através da Porta serial definida.
  - PyAutogui - Biblioteca que executa comandos do mouse e teclado, sendo utilizada para executar as teclas de direção do teclado.
- Virtual Serial Port Driver Trial  
Utilizado para criar uma conexão virtual entre o simulIDE/Arduino e o Python através de portas seriais, permitindo o envio e recebimento de códigos do Simulador para o PySerial.
- Sublime Text 3 - Editor de texto que foi utilizado para desenvolver o código python do projeto.

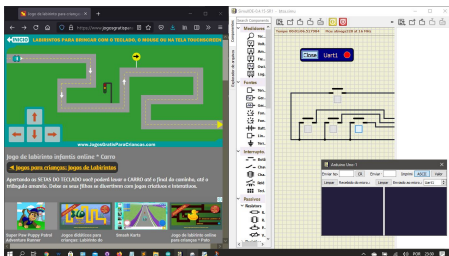
# Arquitetura do Sistema (diagrama de blocos)

Arduino -> SimulIDE -> Virtual Serial Port -> Python -> Execução

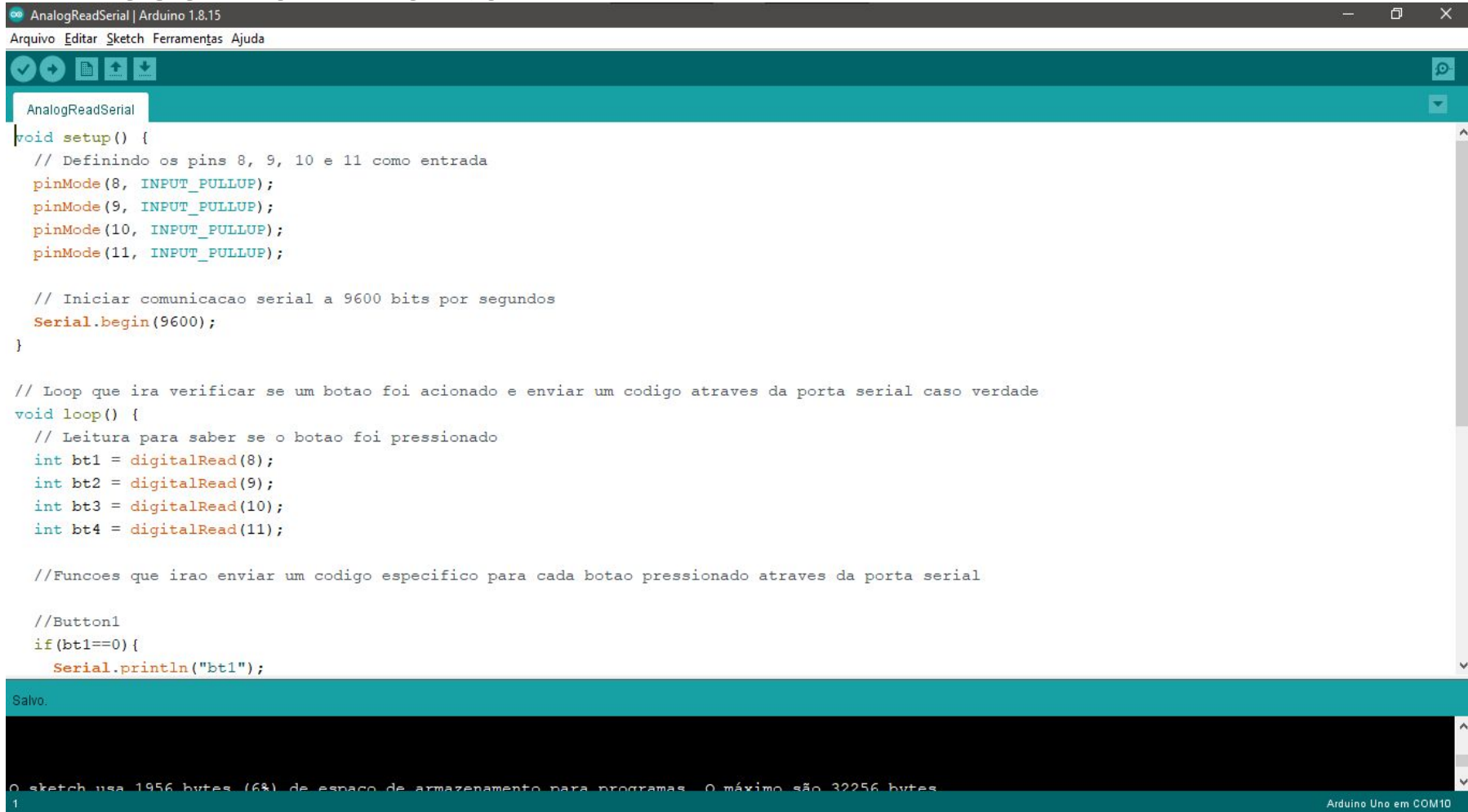
```
1 // Programa para controlar o jogo de labirinto
2
3 // Definição de pinos
4 #define BOTAO 10
5 #define LED 13
6
7 // Definição de constantes
8 const int BOTAO = 10;
9 const int LED = 13;
10
11 // Definição de variáveis
12 int botao = BOTAO;
13 int led = LED;
14
15 // Definição de funções
16 void setup() {
17   pinMode(botao, INPUT);
18   pinMode(led, OUTPUT);
19 }
20
21 void loop() {
22   if (digitalRead(botao) == HIGH) {
23     digitalWrite(led, HIGH);
24   } else {
25     digitalWrite(led, LOW);
26   }
27 }
```



```
1 import serial
2 import time
3 import pyautogui
4
5 # Recebe o código do botão e aciona a tecla respectiva
6 def teclar(bti):
7
8     tecla = ''
9     # Atribui a variável tecla o botão acionado
10    if bti == "bt1":
11        tecla = 'up'
12    elif bti == "bt2":
13        tecla = 'left'
14    elif bti == "bt3":
15        tecla = 'down'
16    elif bti == "bt4":
17        tecla = 'right'
18
19    pyautogui.keyDown(tecla) # Aciona a tecla
20    time.sleep(1)
21    pyautogui.keyUp(tecla) # Desativa a tecla
22
23 # Configuração Porta Serial
24 ser = serial.Serial('COM10', 9600) # alterar para a porta serial disponível
25 time.sleep(1) # tempo necessário para configuração da porta
```



# Desenvolvimento



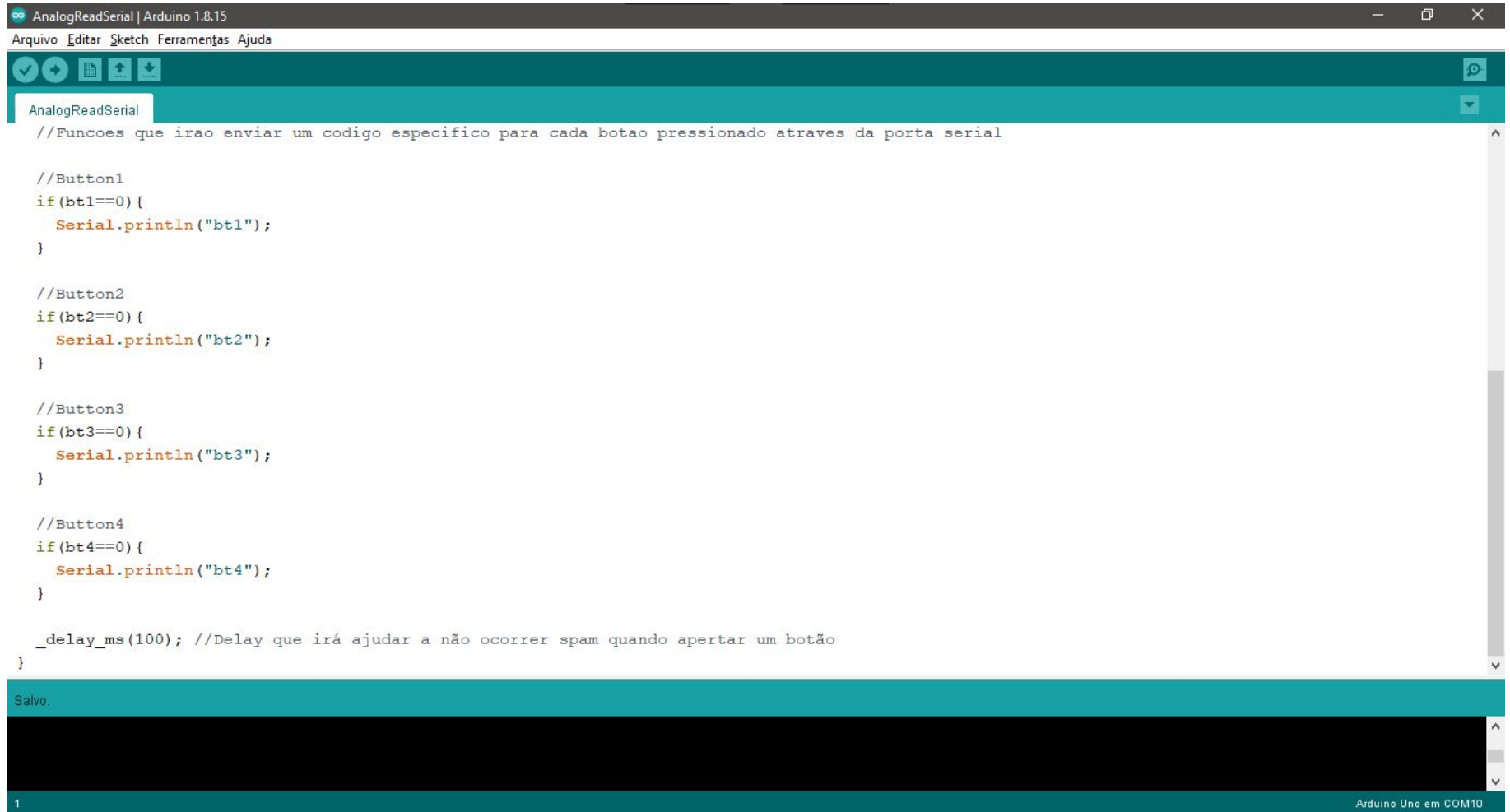
The screenshot shows the Arduino IDE interface. The title bar reads 'AnalogReadSerial | Arduino 1.8.15'. The menu bar includes 'Arquivo', 'Editar', 'Sketch', 'Ferramentas', and 'Ajuda'. The toolbar contains icons for opening, saving, and running. The sketch name 'AnalogReadSerial' is displayed in the top right. The main editor area contains the following C++ code:

```
void setup() {  
  // Definindo os pins 8, 9, 10 e 11 como entrada  
  pinMode(8, INPUT_PULLUP);  
  pinMode(9, INPUT_PULLUP);  
  pinMode(10, INPUT_PULLUP);  
  pinMode(11, INPUT_PULLUP);  
  
  // Iniciar comunicacao serial a 9600 bits por segundos  
  Serial.begin(9600);  
}  
  
// Loop que ira verificar se um botao foi acionado e enviar um codigo atraves da porta serial caso verdade  
void loop() {  
  // Leitura para saber se o botao foi pressionado  
  int bt1 = digitalRead(8);  
  int bt2 = digitalRead(9);  
  int bt3 = digitalRead(10);  
  int bt4 = digitalRead(11);  
  
  //Funcoes que irao enviar um codigo especifico para cada botao pressionado atraves da porta serial  
  
  //Button1  
  if(bt1==0){  
    Serial.println("bt1");  
  }  
}
```

At the bottom, a status bar indicates 'Salvo.' and 'O sketch usa 1956 bytes (6%) de espaço de armazenamento para programas. O máximo são 32256 bytes.' The page number '1' is in the bottom left corner, and 'Arduino Uno em COM10' is in the bottom right corner.



# Desenvolvimento



# Desenvolvimento

```
import serial
import time
import pyautogui

# Recebe o código do botão e aciona a tecla respectiva
def teclar (bti):

    tecla = ''
    # Atribui a variável tecla o botão acionado
    if bti == "bt1":
        tecla = 'up'
    elif bti == "bt2":
        tecla = 'left'
    elif bti == "bt3":
        tecla = 'down'
    elif bti == "bt4":
        tecla = 'right'

    pyautogui.keyDown(tecla) # Aciona a tecla
    time.sleep(1)
    pyautogui.keyUp(tecla) # Desativa a tecla

# Configuração Porta Serial
ser = serial.Serial('COM10', 9600) # alterar para a porta serial disponível
time.sleep(1) # tempo necessário para configuração da porta
```

# Desenvolvimento

```
print("INICIADO")
while 1:

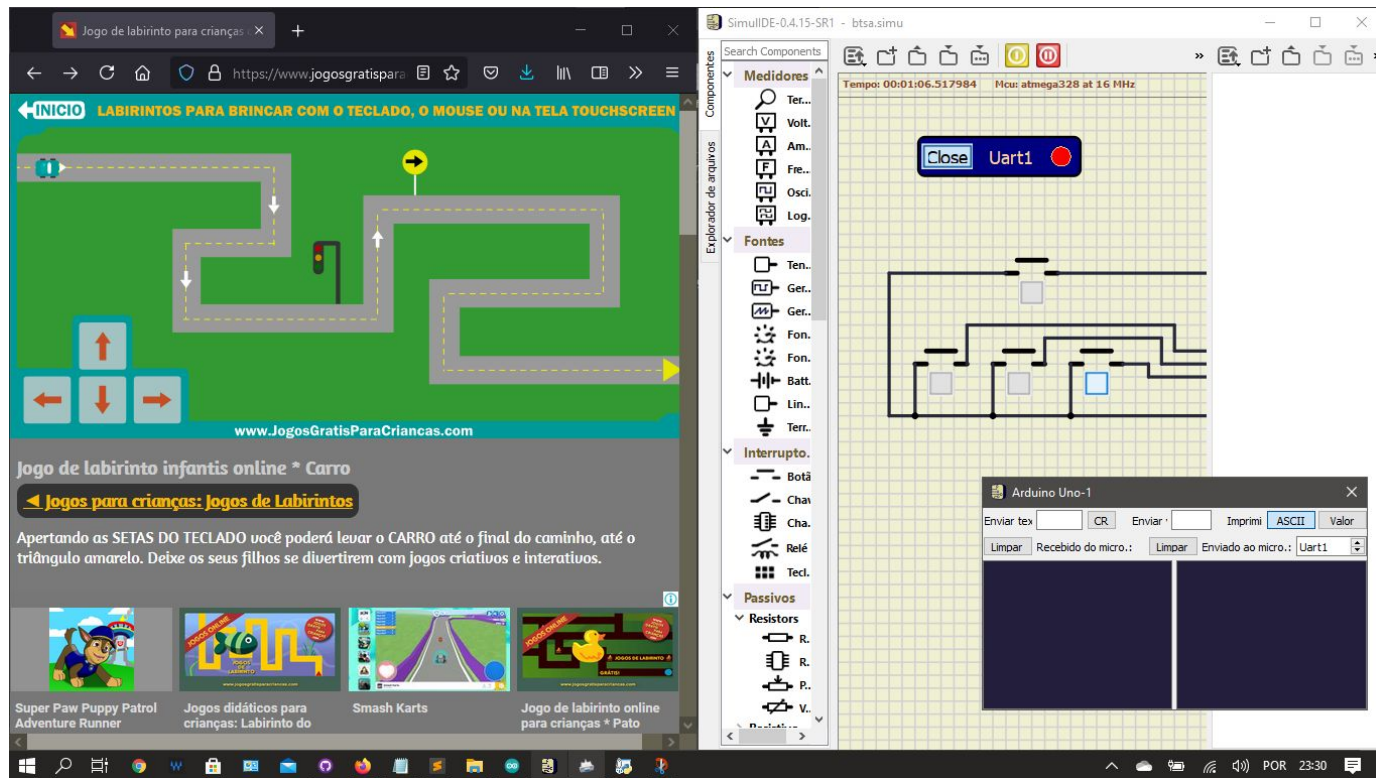
    botao = str(ser.readline().decode().strip('\r\n'))
    print(botao, flush=True)

    #Codigo usado para mover o mouse de janela, mas so esta sendo usado por ser um simulador que aciona o botao e nao usando hardware
    posInit = pyautogui.position() # Posicao Inicial de onde foi feito o clique
    posDest = (200, 200) # Posicao do lado esquerdo da tela
    pyautogui.moveTo(posDest) # Move o mouse para a posicao definida
    pyautogui.click() # Clica com o mouse para deixar a janela da aplicacao ativa

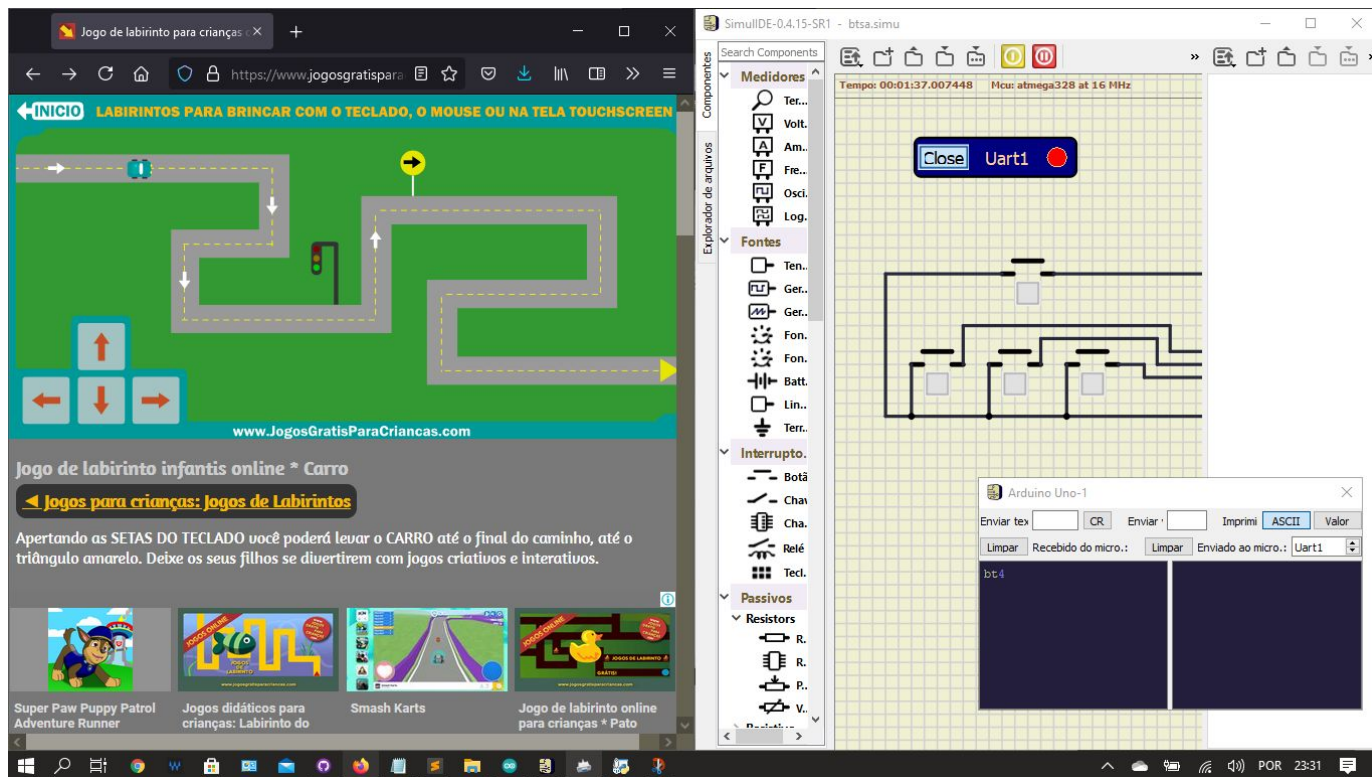
    # Funcao para acionar teclas
    teclar(botao)

    # Retornar mouse para posicao no momento que acionou o botao
    posReturn = posInit + (90,90) # Posicao de retorno
    pyautogui.moveTo(posReturn) # Move o mouse de volta ao simulador
```

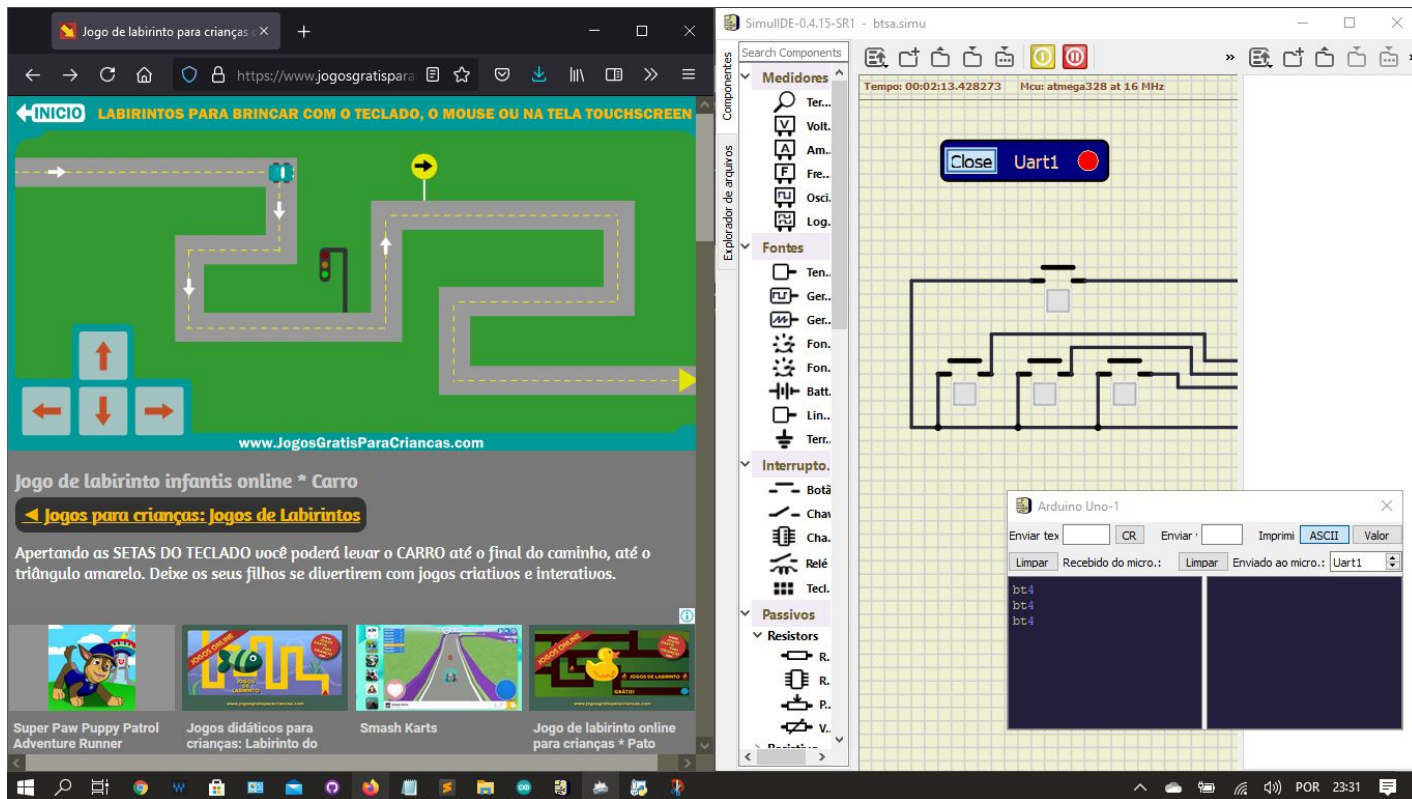
# Resultados



# Resultados

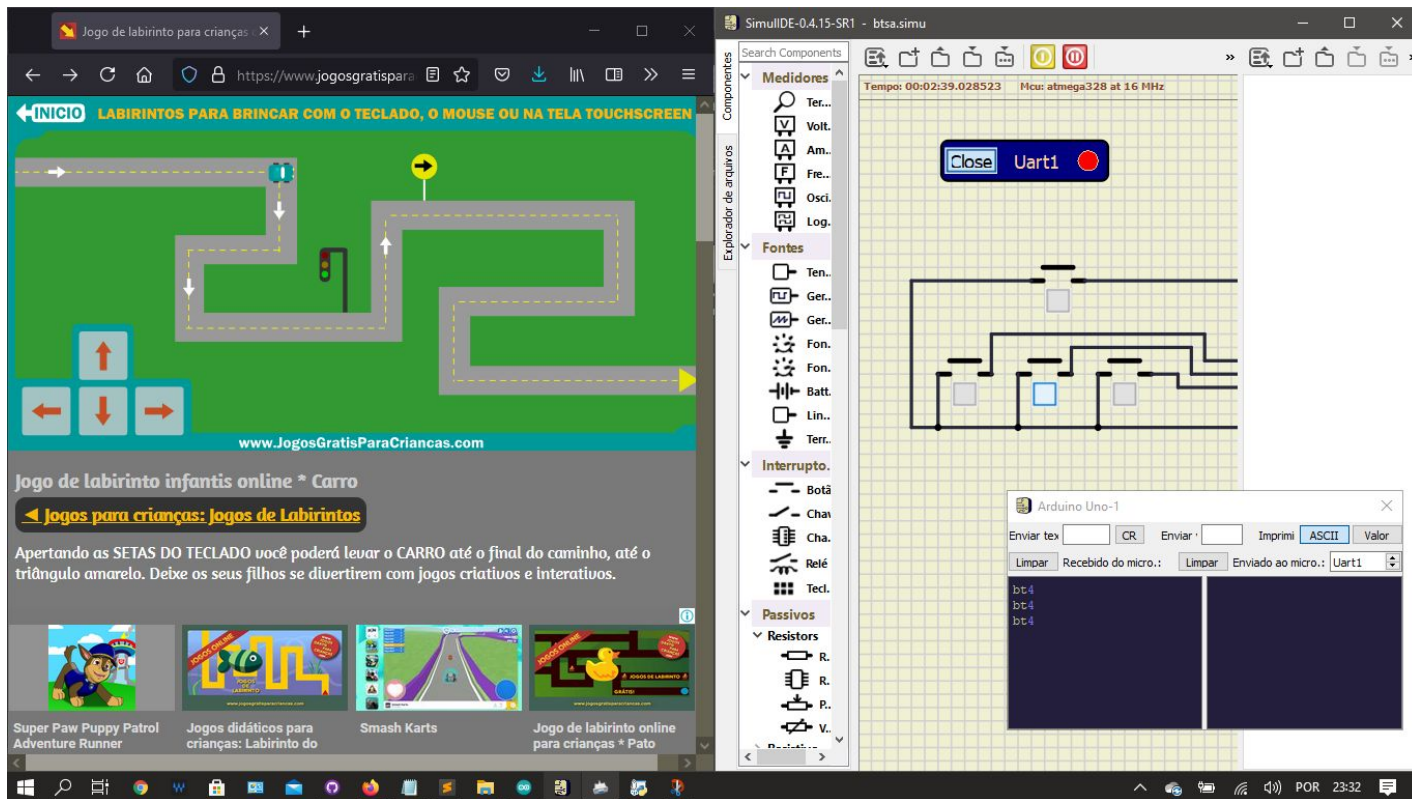


# Resultados





# Resultados



# Resultados

The image displays two side-by-side windows from a Windows desktop.

The left window is a web browser showing the website "JogosGratisParaCrianças.com". The page features a maze game titled "Jogo de labirinto para crianças". The game interface includes a car icon at the start of a dashed yellow path, directional arrow buttons (up, down, left, right), and a yellow triangle at the end of the path. Below the game, there is a section for "Jogos didáticos para crianças: Labirinto do" and several other game thumbnails like "Super Paw Puppy Patrol Adventure Runner" and "Smash Karts".

The right window is a simulation software titled "SimulIDE-0.4.15-SR1 - btsa.simu". It shows a circuit diagram on a grid background. The circuit includes a "Close" button, a "Uart1" component, and several logic gates (AND, OR, NOT) connected in a sequence. A status bar at the top of the simulation window indicates "Tempo: 00:03:44.133911" and "Hcu: atmega328 at 16 MHz". A small window titled "Arduino Uno-1" is open in the foreground, displaying a list of variables: "bt4", "bt4", "bt4", and "bt3". The window also has fields for "Enviar tex", "CR", "Enviar", "Imprimi", "ASCII", and "Valor".



# Conclusão

- Fácil produção  
Com todas estas ferramentas utilizadas, o trabalho ficou bastante viável de desenvolver. O firmware foi o diferencial, fazendo a comunicação do hardware(simulado) com o computador, podendo ser usado como bases para projetos maiores.
- Execução Instantânea  
Por ser um simulador, podemos fazer testes rápidos, verificando falhas e podendo editar uma parte do código para ajustar o projeto.
- Leve  
Utilizando python e arduino, temos um desenvolvimento de firmware bastante leve, sem grandes complicações de verificar se rodará em computadores mais antigos.
- Fácil edição  
Como já mencionado, temos grande facilidade de editar algum algoritmo, ajustando ou acrescentando novas funções.
- Aplicação  
Este projeto poderá ser aplicado de forma real, podendo disponibilizar oportunidades de acesso a tecnologia para pessoas com mobilidade reduzida.

# Referências

## **Xbox Adaptive Controller**

<https://www.xbox.com/pt-BR/accessories/controllers/xbox-adaptive-controller>

## **Teclado Inteligente Multifuncional**

[https://tix.life/categoria\\_aplicacoes/tix/](https://tix.life/categoria_aplicacoes/tix/)

## **Botoeira Usb Cabine Fotográfica Foto Botão Enter Grande**

<https://produto.mercadolivre.com.br/MLB-1209257644-botoeira-usb-cabine-fotografica-foto-botao-enter-grande- JM>

## **Jogo Labirinto**

[https://www.jogosgratisparacrianças.com/labirintos/jogos\\_labirintos\\_carro.php](https://www.jogosgratisparacrianças.com/labirintos/jogos_labirintos_carro.php)

# Anexos

## Projeto

[https://github.com/DanielSenaDev/Sistemas-Embarcados-Codigos/tree/master/BtsA-Botoes\\_Acessiveis](https://github.com/DanielSenaDev/Sistemas-Embarcados-Codigos/tree/master/BtsA-Botoes_Acessiveis)

## Código Python

[https://github.com/DanielSenaDev/Sistemas-Embarcados-Codigos/tree/master/BtsA-Botoes\\_Acessiveis/Python](https://github.com/DanielSenaDev/Sistemas-Embarcados-Codigos/tree/master/BtsA-Botoes_Acessiveis/Python)

## Códigos Arduino

[https://github.com/DanielSenaDev/Sistemas-Embarcados-Codigos/tree/master/BtsA-Botoes\\_Acessiveis/Arduino/AnalogReadSerial](https://github.com/DanielSenaDev/Sistemas-Embarcados-Codigos/tree/master/BtsA-Botoes_Acessiveis/Arduino/AnalogReadSerial)

**Video** (link direto para a pasta video):

[https://github.com/DanielSenaDev/Sistemas-Embarcados-Codigos/tree/master/BtsA-Botoes\\_Acessiveis/Video](https://github.com/DanielSenaDev/Sistemas-Embarcados-Codigos/tree/master/BtsA-Botoes_Acessiveis/Video)