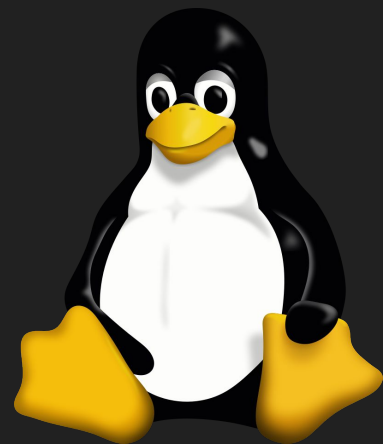


# Projeto de IHS

Transformar um Dispositivo  
Embarcado em Mouse para Linux.

Aluno: Carlos Augusto Santos de  
Carvalho

Professor: Bruno Prado



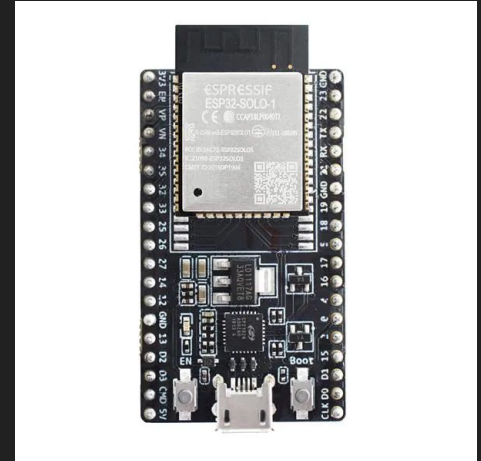
# Ferramentas

Utilizar a biblioteca *uinput*, também conhecida como *libuinput* da linguagem C/C++ permitindo que um programa interaja com o Kernel do Linux como um dispositivo de entrada. Isso significa que um programa pode simular a entrada do usuário, como cliques de mouse ou pressionamentos de tecla, criando um dispositivo virtual com o qual o sistema operacional interage como se fosse um dispositivo físico real.

A biblioteca interage com o módulo do kernel *uinput* através do arquivo */dev/uinput*. Você cria um file descriptor para este dispositivo, configura as capacidades do dispositivo virtual (quais tipos de eventos ele pode gerar), e então envia eventos através de chamadas *write()*.

# O PROJETO

Simular um Mouse através de dois botões conectados a um dispositivo embarcados (ESP32 ou Arduino). Os dois Push Buttons serão colocados em uma Protoboard e conectados à placa via *jumpers*. Ao pressionar um botão, ele irá simular o papel do botão direito de um Mouse convencional. Ao pressionar o outro botão, ele irá simular o funcionamento do botão esquerdo do Mouse.



# Exemplos de Aplicações que a Biblioteca suporta

1. Ferramentas de automação de testes
2. Software de controle remoto
3. Conversores de protocolo (ex: converter comandos de rede em eventos de teclado)
4. Jogos que precisam simular entrada para outros programas
5. Ferramentas de acessibilidade

# Outras Alternativas

Modificar de um driver do tipo **HID (Human Interface Device)** dentro do kernel.

Modificar um **driver existente**, como o *hid-generic* (usado para dispositivos USB HID), ou criar um **novo driver personalizado** para se comunicar diretamente com seu microcontrolador. Compreender como o **Linux trata dispositivos USB HID** e como **criar drivers de dispositivos de entrada**.

O caminho é **criar ou modificar um driver HID** no kernel para que ele **interprete as mensagens do Arduino/ESP32 como cliques e movimentos de um mouse**.

1. **Identificar o dispositivo USB** como um dispositivo HID.
2. **Ler os dados enviados pelo microcontrolador** (via USB, Serial, ou outra interface).
3. **Emular os eventos de um mouse** com base nos dados recebidos, utilizando a API de entrada do kernel.
4. O driver **hid-generic.c** trata todos os dispositivos USB HID genéricos. Se seu dispositivo usa USB HID (como um mouse), ele pode ser tratado diretamente.
5. Se o seu dispositivo não for um **mouse HID** nativo, você pode modificar esse driver ou criar um módulo de kernel que interprete os dados.

# REFERÊNCIAS

<https://kernel.org/doc/html/v4.12/input/uinput.html>