

Proyecto Final: Consola de videojuegos Retro

Fundamentos de Sistemas Embebidos

Reyes Arroyo Pablo Antonio

29 de noviembre de 2024

1. Objetivos

- Consolidar los conocimientos adquiridos durante el curso de fundamentos de sistemas embebidos mediante la creación de un sistema embebido funcional.
- Llevar a cabo la implementación de un centro de entretenimiento que asemeje la emulación de una consola de videojuegos
- Configurar una interfaz inicial que permita la interacción mediante el manejo de un control conectado

2. Material

Se deberá contar de manera vital con la tarjeta Raspberry Pi previamente configurada para la ejecución del centro de videojuegos. Cabe destacar que para el presente proyecto se hizo uso de la Raspberry Pi 5.

- Mando de control Xbox
- Monitor
- Fuente de alimentación USB C 5.1 V a 5 A (
- Cable que permita la conexión entre el mando de control y la tarjeta Raspberry Pi a utilizar (generalmente tipo c-c) y cable que permita la transmisión de video de tarjeta Raspberry Pi a Monitor (tipo c-hdmi)
- Memoria USB con juegos para SNES (opcional)
- Memoria micro SD de capacidad igual y preferentemente mayor a 8GB
- Adaptador para conexión entre micro SD y PC con Windows, Linux o Mac

3. Descripción del funcionamiento de los componentes electrónicos

Desarrollado por Carrasco Servando

Es importante conocer el funcionamiento y las capacidades de los dispositivos empleados, con esto podemos encontrar maneras de aprovecharlos al máximo y conocer si realmente cumplirán con los requerimientos para nuestro proyecto.

3.1. Control de Xbox series

Este componente es fundamental para el proyecto seleccionado puesto que nos permitirá controlar la tarjeta. El control cuenta con los siguientes características:

- Microcontrolador: este componente va a recibir las señales de todos los botones del control y va a traducir las señales para que se envíen a la tarjeta, esto lo va a hacer con ayuda del firmware siendo este último la capa de programación que contenga las instrucciones sobre cómo interpretar cada evento: botones, joysticks y gatillos.
- Botones, joysticks y gatillos: cada uno de estos va a generar una señal eléctrica diferente que se enviará a la tarjeta dado cierto tratamiento por otros componentes.
- Conectividad: un punto importante es cómo se va a transmitir la información, para el caso de este control vamos a tener dos opciones alámbrica, mediante un puerto USB C e inalámbrica, mediante Bluetooth lo que nos brinda más versatilidad.

En resumen, este componente va a recibir eventos cuando presionemos botones, gatillos o movamos joysticks y los va a transmitir a la Raspberry Pi mediante protocolos establecidos por el fabricante.

3.2. Raspberry Pi 5

Un ordenador en toda la extensión de la palabra ya que cuenta con procesador, memoria RAM, puertos de salida y entrada, es importante conocer que la idea básica detrás del proyecto es la de desarrollar una plataforma de computación pequeña y asequible para aumentar el interés por aprender los aspectos fundamentales de las tecnologías de información.[1]

El funcionamiento de la Raspberry es simple y va de la mano con su objetivo: plataforma pequeña de fácil acceso para desarrollo, ya que

- Procesador: Broadcom BCM2712 SoC (System on Chip) A 2.4GHz con cuatro núcleos ARM Cortex-A76. Este componente es el encargado de realizar los cálculos y controlar el flujo de información, recibiendo y ejecutando las instrucciones dadas por el sistema operativo. En pocas palabras, el cerebro de toda la computadora.
- Memoria SDRAM: 2GB, 4GB y 8GB LPDDR4X-4267. Aquí almacenaremos temporalmente la información que será consumida posteriormente por el procesador.
- Conectividad: en cuanto a conexiones inalámbricas contamos con Wi-Fi de doble banda (2.4 y 5GHz), Bluetooth 5.0 y Bluetooth Low Energy, adicionalmente tenemos un puerto Ethernet Gigabit y un total 4 puertos USB, dos 3.0 y dos 2.0 además de un conector GPIO de 40 pines para poder controlar diversos dispositivos como sensores y demás componentes electrónicos, cuenta con dos salidas de video de hasta 4K a 60Hz y finalmente dos conexiones MIPI para cámaras o displays. Todas estas opciones nos permitirán la conexión de periféricos a nuestra tarjeta tanto por medios alámbricos e inalámbricos.[2]
- Almacenamiento: para esto necesitaremos una tarjeta micro SD, también tenemos un puerto PCIe 2.0 con el que podemos conectar también un disco duro SSD M.2, para esto necesitaremos de un adaptador adicional. En estos dispositivos almacenaremos toda la información necesaria para que el sistema se ejecute con normalidad y es la capacidad que tendremos para almacenar cualquier clase de información compatible que queramos.

En cuanto al sistema operativo se prefiere Raspberry Pi OS ya que está adaptado y optimizado para la tarjeta, sin embargo podemos elegir instalar cualquier otro para el que se cumplan los requerimientos del sistema, es importante hacer una elección basada en las necesidades que tengamos.

4. Descripción del funcionamiento de la tarjeta controladora

La tarjeta contronroladora Raspberry Pi, actuará como el núcleo central de lo que corresponde a nuestro sistema embebido, diseñado específicamente para emular una consola de videojuegos. Para esto, se transmitirán datos a través de tres etapas esenciales:

1. Recepción de señales: control y recepción de señales eléctricas provenientes de dispositivos de entrada

2. Procesamiento: Las señales son interpretadas, para que así el software pueda tomar decisiones.
3. Salida: Finalmente, genera una respuesta esperada

Nuestra tarjeta Raspberry Pi tiene cargada una interfaz gráfica diseñada para que usuario tenga la facilidad de poder llevar a cabo la selección del videojuego a ejecutar. Dicha interfaz nos permite establecer una comunicación con el usuario de manera fluida y cumple con las 3 etapas previamente mencionadas:

1. El sistema está configurado para reconocer un mando de Xbox, lo que asegura una compatibilidad específica. Dentro de la interfaz, nuestro sistema solamente reconocerá la pulsación sobre el botón A y movimientos de la cruceta direccional
2. Cuando se detecta una acción que proviene de el mando, el sistema interpretará la señal y ejecuta una tarea específica:
 - a) Desplazamiento: Si la señal se detecta en la cruceta, el puntero de selección se mueve dentro del menú de juegos.
 - b) Confirmación: Si se presiona el botón A, el sistema confirma la selección y ejecuta el videojuego.
3. Nuestro usuario recibirá una respuesta visual y funcional de manera casi inmediata

5. Información sobre el cuidado de la salud y advertencias de riesgos

El uso prolongado de cualquier consola de videojuegos, puede tener implicaciones para la salud si no se toman las precauciones adecuadas [3]. A continuación, se destacan algunos cuidados esenciales y advertencias para minimizar riesgos:

- Cuidado de la vista: es importante jugar en una habitación que cuente con una buena iluminación y mantener una distancia adecuada de la pantalla
- Ergonomía y postura: Se recomienda utilizar sillas ergonómicas y colocar la pantalla a la altura de los ojos para evitar tensiones musculares (principalmente cuello y espalda).
- Descansos regulares: Se aconseja tomar pausas de 10 a 15 minutos cada hora de juego continuo. Esto ayuda a reducir la fatiga ocular y el cansancio muscular
- Sobrecalentamiento de Raspberry: Se deberá colocar la tarjeta Raspberry Pi en un lugar ventilado y estable para evitar el sobrecalentamiento. Es esencial no cubrir la placa.
- Riesgos eléctricos: Verificar que la fuente de alimentación sea la adecuada (5V/5A) y que los cables estén en buen estado reduce el riesgo de cortocircuitos o incendios.
- Advertencia sobre la epilepsia fotosensible: Se recomienda que cualquier usuario con antecedentes de epilepsia consulte a un médico antes de usar la consola.

6. Información sobre el cuidado de los componentes electrónicos delicados

Desarrollado por Carrasco Servando

- Cuidado en la manipulación de los equipos: debemos tener cuidado de evitar caídas y golpes fuertes en los dispositivos, puede ocasionar fallos en el hardware, en general, debemos evitar manejos bruscos de los equipos, y de igual manera debemos evitar el contacto con agua ya que podría dejar nuestros dispositivos inservibles, por último hay tener cuidado de las descargas electrostáticas, usando una pulsera antiestática de ser posible y siempre trabajar sobre una superficie antiestática, más aun si nuestra Raspberry no cuenta con un case protector siempre es recomendable conseguir uno para mayor protección. Todo lo anterior también aplica para los elementos que van conectados a la Raspberry como lo puede ser la tarjeta micro SD.

- La Raspberry puede sobrecalentarse si se usa intensivamente, es por esto que se recomienda algún tipo de disipador y de ser posible un pequeño ventilador y una posición adecuada que permita el flujo de aire fresco.
- Actualizaciones: debemos tratar de mantener nuestros equipos actualizados en la medida de lo posible, si bien sabemos que estos sistemas llevan años en el mercado aún puede haber fallas por descubrir, mantener el equipo actualizado puede mantener nuestros equipos funcionando correctamente y corrigiendo errores.

7. Configuración de la tarjeta controladora

Desarrollado por Carrasco Servando

Necesitaremos de diversas configuraciones para lograr que nuestra tarjeta de video logre ejecutar el programa de este proyecto y logre emular a una consola.

7.1. Sistema operativo

Vamos a necesitar de Raspberry Pi OS lite de 64 bits específicamente la versión del 13 de noviembre del 2024, este viene sin entorno gráfico, por lo que será más fácil de manipular, pesará menos que otras distribuciones con escritorio y tendremos un nivel de control más alto. Para instalar Raspberry Pi OS lite necesitaremos realizar los siguientes pasos:

1. Descargar [Raspberry Pi OS lite](#) y [Raspberry Pi Imager](#) de la página oficial.
2. Abrir e instalar Raspberry Pi Imager y seleccionar la imagen del sistema previamente descargada, luego debemos introducir nuestra tarjeta micro SD a la computadora y seleccionar la tarjeta en el programa.
3. Escribir la imagen en la tarjeta y confirmar el formateo de la tarjeta micro SD.
4. Cuando finalice podemos introducir la tarjeta a la Raspberry alimentarla y encenderla.
5. Realizar la configuración inicial de la tarjeta, nombre de usuario y contraseña, es importante recordar que para esto necesitaremos un teclado, ya que este sistema se maneja solo por comandos de terminal.
6. Con el comando `sudo raspi-config` podremos conectarnos a una red, tendremos que ir a **System Options > Wireless LAN** ingresar el nombre de la red y la contraseña, una vez hecho esto estaremos conectados a internet.
7. (Opcional) si queremos habilitar el inicio de sesión automático tendremos que ir a **Boot / Auto Login > Console Autologin** con esto la Raspberry iniciará sesión por sí sola.

7.2. Librerías necesarias

Para la correcta ejecución de los programas que descargaremos son necesarias algunas librerías, las librerías son conjuntos de funciones, definiciones y demás recursos que se pueden reutilizar y que son requeridas por ciertos programas.^[4]

La forma en la que estas librerías se instalan es siguiendo la siguiente estructura:

```
$ sudo apt install nombre-librería nombre-librería2 ...
```

- libSDL2-dev: Soporte para gráficos, sonido y entradas. [\[Más información\]](#)
- libglu1-mesa-dev: Implementación de interfaz OpenGL. [\[Más información\]](#)
- libasound2-dev: Salida de audio. [\[Más información\]](#)
- libpng-dev: Necesaria para la gestión de imágenes PNG. [\[Más información\]](#)
- libjpeg-dev: Necesaria para la gestión de imágenes JPEG. [\[Más información\]](#)
- libudev-dev: Soporte de dispositivos. [\[Más información\]](#)

- libzip-dev: Soporte de archivos zip [\[Más información\]](#)
- git: controlador de versiones. [\[Más información\]](#)
- xorg: manejar interfaz gráfica.

7.3. Librerías requeridas por Python

Similar a las mencionadas anteriormente, estas librerías son requeridas por el lenguaje de programación Python, en el cual desarrollamos nuestro programa principal. La forma de instalarlas es similar a la estructura ya mencionada:

```
$ sudo apt install python3-librería python3-librería2
```

- python3-tk: Soporte para interfaces gráficas en Python. [\[Más información\]](#)
- python3-evdev: Manejo de dispositivos de entrada en Python. [\[Más información\]](#)
- vlc: Reproductor multimedia. [\[Más información\]](#)
- python3-vlc: Enlace de Python para VLC. [\[Más información\]](#)
- python3-pyudev: Interacción con Udev en Python. [\[Más información\]](#)

7.4. Instalación del emulador

Para descargar, compilar y ejecutar el emulador haremos lo siguiente:

- Abrir una terminal y ejecutar el siguiente comando para descargar el código fuente:
\$ `wget https://mednafen.github.io/releases/files/mednafen-1.32.1.tar.xz`
- Descomprimir el archivo con el siguiente comando:
\$ `tar -xvf mednafen-nombre.tar.xz`
- Entrar a la carpeta de instalación con \$ `cd mednafen/` dentro de esta carpeta ejecutaremos el siguiente comando para preparar el código fuente para su posterior compilación: \$ `./configure`
- Compilar el código fuente, primero necesitamos entrar a la carpeta generada src \$ `cd src/` y con \$ `make`, vamos a empezar la compilación este proceso puede ser tardado.
- Instalar el emulador, \$ `sudo make install`
- Copiar el archivo de configuración `mednafen.cfg` a la ruta de instalación de mednafen, esto lo haremos con el siguiente comando: \$ `cp mednafen.cfg /home/usuario/.mednafen/`, este archivo contiene la configuración para que los controles de Xbox sean detectados y se puedan usar para jugar.

Cuando hayamos hecho esto podremos llamar al emulador con el comando \$ `mednafen`, si quisiéramos iniciar un juego tenemos que ejecutar \$ `mednafen roms/mario.sfc`, suponiendo que tenemos un directorio llamado roms y dentro de este se encuentra un juego.

Para ejecutar nuestro programa primero tenemos que pasarlo a nuestro dispositivo y ejecutar el siguiente comando estando en la ruta `"/home/usuario/"`:

```
$ sudo echo "python3 final.py" > .xinitrc
```

con esto crearemos un archivo `.xinitrc` (oculto) para que al iniciar el entorno gráfico, este ejecute nuestro programa, con esto solo bastaría con ejecutar el comando: \$ `startx` y listo, nuestro emulador con interfaz estará ejecutándose.

7.5. Algunas consideraciones

- Si alguno de los comandos nos arroja el problema de no tener permisos o acceso denegado, simplemente debemos agregar la palabra `sudo` al inicio del comando, tal vez puede pedirnos nuestra contraseña de usuario, esto quiere decir que estamos ejecutando el comando con permisos elevados.
- Si deseamos que el emulador se abra tan pronto como encendamos la Raspberry necesitaremos ejecutar el siguiente comando: `$ echo "startx" > init_console.sh` y luego:
`$ echo "/home/usuario/init_console.sh" >> .bashrc` lo que estamos haciendo con estos comandos es crear un archivo `init_shel.sh` que se ejecute al iniciar la Raspberry, con el comando `startx`.
- Los juegos soportados, para la consola SNES son los que tienen formato `.sfc` y `.smc`, la manera legal de conseguirlos es comprando los juegos originales y extrayendo las copias para preservación.
- Para más información del emulador tenemos la [\[documentación\]](#)
- El repositorio que contiene todos los archivos mencionados es el siguiente: [\[repositorio en GitHub\]](#)
- El video del funcionamiento del proyecto lo podemos revisar [aquí](#)

8. Desarrollo de los componentes de software

Para cumplir con la tarea principal de nuestro sistema, se llevó a cabo la creación de 6 módulos de software, los cuales cada uno cumple una tarea y se ejecutarán dentro de 3 procesos. Dichos módulos son los siguientes:

- Módulo de carga de video:

La finalidad de este apartado, es el llevar a cabo la búsqueda y ejecución del video que corresponde al encendido de la consola. Esto se lleva a cabo mediante el uso de la librería `vlc`, por lo que inicia con una instanciación y posteriormente se lleva a cabo la asignación del archivo, para finalmente poder ejecutar el video y transcurridos 7 segundos, se detendrá la reproducción del video.

```
def load_video():
    player = vlc.MediaPlayer()
    video = vlc.Media("/home/user/video.mp4")
    player.set_media(video)
    player.play()
    time.sleep(7)
    player.stop()
```

- Módulo de carga de videojuegos

En la función que lleva por nombre `load_videogames()`, lleva a cabo lo que corresponde a la creación de un diccionario que contendrá los archivos que sean de tipo `.smc` o `.sfc` y se encuentren ubicados en la ruta `/home/user/roms/`. Esto nos permite llevar a cabo la carga de videojuegos de manera dinámica y de acuerdo a los recursos con los que cuenta el usuario.

- Módulos de comandos

Se contienen los comandos que se encargarán de ejecutar la instrucción deseada por medio de instrucciones del sistema operativo

- Módulo de indentificación de USB

Como su nombre lo indica, hace referencia las funciones: `auto_mount(path)`, `get_mount_point(path)`, `file.exists_in_destination(file, destination_dir)`, `move_files(source_dir, destination_dir)` y `listen_for_usb()`. Estas 6 funciones, a grandes rasgos, cumplen con la tarea que se debe desempeñar al detectar un nuevo dispositivo conectado, empezando por la detección, seguido de un automontaje, y por último el copiado de archivos llevando a cabo la identificación de éstos para no llevar a cabo la creación de duplicados.

- Módulo de interfaz de gráfica

Se encarga de llevar a cabo la creación de la interfaz gráfica que servirá como herramienta de comunicación con el usuario para poder llevar a cabo la tarea principal del sistema. En este apartado se lleva a cabo la creación de la ventana, su contenido como el listado de videojuegos, los botones para apagar y reiniciar consola y de igual manera intera las funciones que conllevan las tareas encargadas de actualizar dicha interfaz para que el usuario pueda conocer que su acción fue detectada correctamente.

- Módulo de identificación de mando de control

Nuestro último módulo tiene la finalidad de identificar y escuchar los eventos que se reciben desde el mando de control conectado a nuestra consola. Como primer paso, se deberá filtrar los posibles dispositivos conectados tomando como criterio la palabra "xbox.^{en} el nombre del dispositivo. Posterior a eso, se tiene la identificación de los botones principales que serán de ayuda para llevar a cabo alguna tarea específica dentro de la interfaz gráfica, como puede ser el desplazamiento sobre el listado de videojuegos, selección, apagado y reencendido de la consola.

9. Integración de los componentes de software

Debido a que se trata de un sistema donde cada módulo es dependiente de 1 o más módulos, se optó por unificar en un solo archivo los módulos creados. Esto con la intención de facilitar el flujo de trabajo. Al iniciar la ejecución del archivo final.py, se deberá cargar el video para llevar a cabo la simulación de inicio de consola, lo cual añade un grado de realismo mayor a nuestra interfaz. Posterior a eso, se hace la creación de un hilo secundario, el cual tendrá como tareas las que corresponden a la configuración de dispositivos USB, es decir, se encargará de actuar como el listener monitoreando continuamente las conexiones de dispositivos externos. Si detecta la presencia de un dispositivo nuevo, verifica si contiene videojuegos y, de ser necesario, copia los archivos faltantes al directorio local para garantizar que estén disponibles en el sistema. Por último, se ejecutará el proceso de inicializar la carga de la interfaz gráfica donde, como ya se hizo mención, se realizará la creación y configuración de ésta. Para el manejo del control, se manda a instanciar un nuevo hilo dentro del proceso de interfaz gráfica, reflejando la dependencia directa entre ambos elementos: sin una interfaz correctamente inicializada, el control de mando no tendría sentido, ya que no habría un entorno en el que aplicar las señales recibidas.

10. Conclusiones

La elaboración del presente proyecto, me permitió obtener un panorama mucho más grande sobre lo que conlleva el desarrollo de sistemas embebidos, los cuales están presentes en el día a día. Fue una parte fundamental el buscar soluciones para poder completar el presente proyecto en las que se requería la modificación de ciertos archivos del sistema. De igual manera, el lograr desarrollar un centro de entrenamiento que emula una consola requiere un amplio análisis sobre el como identificar señales que provienen de equipos externos, los cuales no serán los default. Por último, el lograr implementar la interfaz gráfica es de gran ayuda y aunque no se tienen los mejores gráficos, se puede desarrollar algo similar con el objetivo que era desarrollar una consola retro.

11. Cuestionario

- Explique ¿qué es un emulador?

Tomando como referencia a la definición de [IBM](#), "Un emulador es una aplicación de software que permite al sistema funcionar como si se estuviera utilizando un terminal o una impresora diferente.", por lo que podemos definir a un emulador a un software que nos permite ejecutar tareas y/o procesos sobre un sistema para el que no fue diseñado pero cumpliendo los requerimientos. [5]

- Es legal la emulación.

La emulación no es ilegal si se desarrolla con métodos propios y no infringiendo en derechos de autor, lo que puede llegar a ser ilegal es la obtención de juegos si no los obtenemos de copias originales.[6]

- Explique las ventajas y desventajas que tiene utilizar un emulador.

Ventajas: Permiten utilizar sistemas que de otro modo no podríamos usar, proporcionan facilidades al programador de estos sistemas para el desarrollo de aplicaciones, generar documentación para futuras reimplementaciones, flexibilidad y portabilidad de aplicaciones, ahorro de costos en equipos necesarios y/o dedicados.

Desventajas:El rendimiento se puede ver directamente afectado, se pueden observar una cierta cantidad de problemas con la compatibilidad, algunos emuladores requieren de ciertos requerimientos de hardware para no generar afectaciones al usuario. [5]

- Ventajas y desventajas de mednafen.

Una de las principales ventajas es la posibilidad de emular varias plataformas como lo son NES, SNES y GameBoy Advance, además de su fácil instalación. Como desventaja podría ser la configuración, el archivo de configuración puede ser confuso, adicional a esto, puede llegar a haber problemas con el audio, teniendo que pasar el driver de audio por comando a la hora de ejecutar el emulador.

- La Raspberry solo sirve para un propósito específico?

No, como se mencionó en el desarrollo del proyecto, la capacidad de instalarle distribuciones de Linux con escritorio puede convertir a la pequeña Raspberry Pi en un ordenador personal con potencia suficiente para tareas básicas.

- ¿Es posible usar el control de manera inalámbrica?

Nuestra tarjeta Raspberry Pi cuenta con módulo de bluetooth, por lo que podemos llevar a cabo la configuración que permita el emparejamiento entre la tarjeta y el mando de control.

12. Bibliografía de consulta

Referencias

- [1] Derek Molloy. *Raspberry Pi a fondo para desarrolladores*. Marcombo, 2019.
- [2] Patricia Quiroga. *Arquitectura de computadoras*. Alpha Editorial, 2010.
- [3] Byron C. “El impacto de los videojuegos en la salud de las personas: una mirada crítica”. En: (2023).
- [4] Christopher Negus. *Linux Bible*. Wiley, 2020.
- [5] B. Farrand. “La emulación es la forma de adulación más sincera: Videojuegos retro, distribución de ROM y derechos de autor”. En: *IDP. Revista de Internet, Derecho y Política* (2012).
- [6] Miquel Térmens. *Preservación digital*. UOC, 2010.