Proyecto Final: Centro Multimedia

Oropeza Sanchez Guadalupe Monserrat NL: 22

Noviembre 26, 2024

1. Introduction

An embedded system is a microprocessor-based system that is built to control one or more functions and is not designed to be programmed by the end user in the same way as, for example, a Personal Computer.

An embedded system is a controller that sits inside a larger system to perform a dedicated function; they are used in a large number of modern devices, including household machines: microwaves, toasters and washing machines.

The Raspberry Pi is a single-board computer that is similar in size to a computer mouse (85 mm x 56 mm is its standard size). Visually, it resembles a PC motherboard, but includes all the essential integrated components, such as the processor (CPU), memory, wireless module, USB ports and network connection. However, it does not have integrated storage, such as a hard disk. Instead, it has a micro-SD card slot on the back and supports external USB drives to expand its storage capacity. A media center is software that allows you to manage and enjoy all your multimedia content; it can play music, movies, display images, and so on. These media centers can run on a mobile device, on a PC or, in this case, on a Raspberry Pi.

For the development of this multimedia center, Tkinter was used, it works for the creation and development of desktop applications, this library facilitates the positioning and development of a desktop graphical interface.

2. Objectives

The objective of this project is to develop an embedded entertainment system that allows users to play movies, videos, music and photos both from removable media and through selected online services. This system is designed to offer an intuitive and seamless experience, ensuring compatibility with modern multimedia content and prioritizing an accessible and easy-to-use interface.

This project represents a technical challenge that requires the skills acquired during the embedded systems course, it is a relevant project as it addresses a real user need.

3. List of materials

- Raspberry Pi 4.
- Power supply for Raspberry Pi.
- Micro HDMI to VGA cable or micro HDMI to HDMI adapter.
- MicroSD card 8 GB (minimum).
- USB drive with multimedia content.

- Monitor.
- Keyboard.
- Mouse.
- Speakers or headphones 3.5 mm.

4. Description of components operation

4.0.1. Micro HDMI to VGA cable

- Video Transmission: This cable allows you to connect Raspberry Pi to a monitor with VGA input, facilitating high quality video output. It is a practical solution when the monitor does not have HDMI input.

4.0.2. USB drive

- File Storage: The USB drive employs non-volatile memory technology, ensuring data retention without the need for constant power. It is ideal for storing and transferring multimedia content. Connection: The USB interface enables fast and efficient data transfer between Raspberry Pi and the external storage device.

4.0.3. Keyboard and Mouse

- Control and Interaction: These peripherals are used for navigation through the system interface and control of its functions. They are connected via the USB ports of the Raspberry Pi, ensuring immediate response.

4.0.4. Monitor

- Output Screen: It is the device in charge of displaying the images and videos processed by Raspberry Pi. Its function is fundamental for the visual interaction with the system.
- Adapter Compatibility: It can be connected via HDMI cables or micro HDMI to VGA adapters to ensure compatibility with different models and monitor technologies.

4.0.5. Power Supply

- Stable Power: Provides the power supply necessary for proper operation of the Raspberry Pi. An original power supply of 5V and at least 3A is recommended to avoid power failures and ensure system stability.

4.0.6. Speakers and Headphones

- Audio Output: This component is connected via a 3.5mm audio jack to provide clear and quality sound for multimedia playback. It is an essential element to fully enjoy the listening experience of the system.

4.0.7. microSD card

- Boot System: Acts as the main disk where the operating system (Raspbian Lite) and the project's core files are stored. It allows the system startup and the execution of multimedia applications.

5. Health care information

To ensure user safety, direct contact with the Raspberry Pi when it is in operation should be avoided, as some components may become hot and may cause minor burns. We should ensure that we work in a well-lit and ventilated space to reduce eye strain.

6. Component care information

It is important to store electronic components in a dry, dust-free place to avoid moisture or small particles. When connecting or disconnecting devices such as the Raspberry Pi or monitor, always with the system powered off to prevent short circuits. Also, handle the Raspberry Pi using an antistatic surface to avoid damage from static electricity, and avoid placing heavy objects on the cables or the board, as they could be bent or damaged.

7. Board configuration

.

For proper operation of the Raspberry Pi, it is necessary to connect all the required peripherals, such as the keyboard, mouse and monitor, making sure to use the appropriate cables. Once the peripherals are ready, turn on the Raspberry Pi by connecting to a power supply. In addition, it is essential to prepare the Raspberry Pi operating system. Raspberry Pi OS Lite must be downloaded and installed on a microSD card, this would be the physical specifications to run the multimedia system. In addition, certain requirements are needed for the media center to work:

- Tkinter
- Pyudev
- VLC
- Xorg
- Chromium
- Pulseaudio

8. Module development

Main module MAIN.PY

Acts as the entry point for the media center, initializes the main window and manages navigation between the different interfaces.

Details: beginitemize is used, window dimensions and properties are set.

Switches between interfaces (network, streaming, USB)

Network module NETWORK.PY It is responsible for managing and connecting to WiFi networks. Details

- The available networks are scanned and displayed in a list, we can select a network and enter the password of that network. connections are displayed and updated depending on their availability.

  Streaming Module STREAMING.PY Online streaming services can be accessed. Details
- You can access to the streaming services through buttons.
- Each button opens the browser in Chromium and displays the full tab.

USB module USB.PY Manages the playback of multimedia content from connected USB devices.

- Monitors connected USB devices, creates dynamic buttons for photos, videos and music.
- Runs a separate thread to detect USB devices and update the interface in real time.

9. module integration

- MAIN MODULE

This module acts as the core and entry point of the media center application. It configures the main window using Tkinter, defining its size and properties, and manages the navigation between the different user interfaces, such as network, streaming and USB, using specific methods that clean and load the corresponding interface.

- NETWORK MODULE Responsible for the configuration and management of the WiFi network, this module contains the class `NetworkInterface`, which allows to scan available networks, display a list of options, select a network and enter a password to connect. It uses `WifiManager` to perform these operations and update the connection status on the interface. This module was tested not only with the networks we had range with, we also created access points with our cell phones to check that it could detect them.
- STREAMING MODULE This module manages streaming services such as Netflix, prime video, Spotify, etc. The buttons of each of the applications are presented, and with Chromium in application mode the content is played in full screen, and all applications have a back button to return to the streaming screen.
- USB MODULE

  In this module we have the reproduction of multimedia content, when connecting the USB, it recognizes the memory and analyzes the elements that it contains, whether audio, images or video, the images are reproduced in a presentation, changing each one every 5 seconds, it shows us a list of videos or the option to reproduce them in presentation, and with the audio, it allows us to reproduce it, it has the buttons, to pause, resume, next, previous and return to the main menu, in the audio part a timer was used for the reproduction of songs.

10. Conclusions

With this project we learned how to assemble and program on a Raspberry Pi, something we did during the whole semester, but it was taken to a more advanced level, all the concepts seen in the lab and theory were used and embodied in this project.

The difficulties that we had in this project first was the development of the interface, we started doing it with PyGames but we saw that this was not going to fulfill what we needed, because to change the interface had to happen an event, then we did it with Tkinter, which was much simpler.

When we needed to resize the image to fit the screen on which it was being played, the icons were out of sync and the appearance of our interface changed.

Another difficulty was that our program ran in unattended mode, since when the permissions were configured to do it this way, the audio was involved and it stopped working in streaming, audio and video.

Something that we failed to implement in this multimedia center was the control command, in our planning we wanted to do it with an IR control and IR sensor, but at the time of integrating it into our system it failed, and we chose to manage it with keyboard or mouse.

Referencias

[1] PulseAudio Developers. *PulseAudio Documentation*. https://www.freedesktop.org/wiki/Software/PulseAudio/. Accessed: 2024-11-26. 2024.

[2] screeninfo Developers. *screeninfo: Get monitor information from the OS*. https://github.com/rr-/screeninfo. Accessed: 2024-11-26. 2024.

[3] FFmpeg Team. *About FFmpeg*. Accedido en 26 de noviembre de 2024. 2024. URL: https://www.ffmpeg.org/about.html.

[4] Python Software Foundation. *Tkinter — Python Interface to Tcl/Tk*. https://docs.python.org/3/library/tkinter.html. Accessed: 2024-11-26. 2024.

[5] G. Galeano. *Programación de sistemas embebidos en C*. Colección 3B. México: Alpha Editorial, 2009.

[6] Google LLC. *Widevine CDM*. https://www.widevine.com/. Accessed: 2024-11-26. 2024.

[7] Fredrik Lundh. *An Introduction to Tkinter*. Accedido en 26 de noviembre de 2024. 1999. URL: http://www.pythonware.com/library/tkinter/introduction/index.htm.

[8] D. Molloy. *Raspberry Pi® a fondo para desarrolladores*. Marcombo, 2019.

[9] VideoLAN Organization. *VLC Media Player*. https://www.videolan.org/doc/. Accessed: 2024-11-26. 2024.

[10] The Chromium Projects. *Chromium Browser*. https://www.chromium.org/. Accessed: 2024-11-26. 2024.

[11] Python Imaging Library Team. *Pillow Documentation*. https://pillow.readthedocs.io/en/stable/. Accessed: 2024-11-26. 2024.

[12] pyudev Team. *pyudev — libudev bindings for Python*. https://pyudev.readthedocs.io/en/latest/. Accessed: 2024-11-26. 2024.

11. Adicional

YouTube test video: https://youtu.be/JCH2hj2dtEo?si=ZdE_1C7KUk0pxib1

Repository: https://github.com/m4rc0m0nt3l0ng0/Proyecto_Final_Centro-Multimedia.git