Final Project: MaFe TV

Marco Antonio Piña Rossette

May 21, 2024

1 Introduction

An integrated or embedded system is a controller inside a bigger system with an specific dedicated function. Embedded systems are used in modern devices, including domestic machines like the microwave, toasters and washing machines.

The complexity of an embedded system can vary significantly depending on the task it is designed for, ranging from a single microconttroller to a set of chips with connected peripherals and networks. [1]

A media center, or multimedia center, is software that allows you to have and enjoy all your multimedia content. It can play music, movies, display images in galleries, etc. All from multimedia contet stored on a local hard drive, or accessing this content through the network (...) These media enters can be running on a mobile device, on a PC, or on a Raspberry Pi.

2 Objective

The objective of this project is to implement an embedded entertainment center that allows users to play movies, videos, music, and photographs from both removable media and online services, in other words, a multimedia entertainment center.

This project is a technical challenge that requires students to develop their skills in embedded systems development and multimedia playback. The project is also relevant and significant, as it addresses a real user need.

3 Materials

- Raspberry Pi 3
- Raspberry Pi's power supply
- HDMI cable
- Micro SD card (8 GB)
- USB with multimedia content
- Monitor (TV)
- Keyboard
- Mouse
- Speakers

4 Electrical components description

- **USB memory:** It is used to save data, in the form of pictures, videos, music or movies.
- **Monitor:** It's used as the output device to show images and videos.
- **Speakers:** It's used to play the sound of the multimedia content.

5 Health Care Information and Risk Warnings

When working with the Raspberry Pi 3 card, it is important to take certain precautions, however, it does not pose a major risk to people's health.

Sometimes the card can get hot from use, so it is recommended not to touch it when it is in use, it could cause a minor burn.
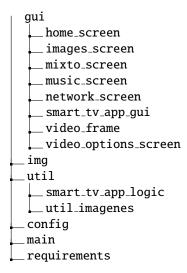
Whenever the Raspberry Pi is plugged in and in use, it is recommended not to touch it, as it is plugged into electrical power and could give mild shocks.

6 Controller device description

The microcontroller works by connecting it to a monitor or TV and after being powered, it starts with the graphical interface of the MaFE TV system. The raspberry pi will automatically detect the display and speakers that the system will use.

7 Software modules description

The MaFE TV is divided into two main modules: the Graphical User Interface (GUI) located in the `gui` directory, and the app logic, located in the `util` directory. The directory tree has the next configuration.

```
gui
    home_screen
    images_screen
    mixto_screen
    music_screen
    network_screen
    smart_tv_app_gui
    video_frame
    video_options_screen
img
util
    smart_tv_app_logic
    util_imagenes
config
main
requirements
```

7.1 GUI

The grpahical interface works by having singular panels to draw in the `smart_tv_app_gui.py` file, those are a top panel with the author's information and the project name, the side panel that is used as a side menu, and the central panel in which we draw all the other screens in the `gui` directory.

7.2 Logic

All the logic behind the app is located in the `util/smart_tv_app_logic.py` file. The class in that file is instantiated as an attribute of the SmartTVAppGUI class to use the methods that allow the app to do things.

7.3 Streaming functionality

The streaming functionality is implemented using the choromium browser in mode app. This allows the project to run streaming services like Netflix, Spotify and Youtube. It also allows the project to run Google. In the GUI this is the

HomeScreen class.

7.4 USB detection

To detect when an external hard drive is connected, the MaFE TV system runs a thread in deamon mode and uses pyudev to monitor the connection of external drives. When it detects one, uses the logic module to mount and analyze the content of the USB to do an action, that can be reproducing the music in infinite loop using VLC, reproducing the selected video using VLC, showing the images in presentation mode using Pillow. Depenting on the content of the usb, in the GUI this can be the **ImagesScreen**, **VideoOptions**, **MusicScreen**, or **MixtoScreen** class.

7.5 Network Configuration

To allow the user to connect to the internet, we made a network assistant that displays the available networks and allows the user to enter a password and connect to the selected SSID. This works using the Linux Network Manager CLI. In the GUI this is the **NetworkScreen** class.

7.6 Software components integration

The SmartTVAppGUI instantiates the SmartTVAppLogic to use the methods that allow the application to function, SmartTVAppGUI changes the screen to show what the user wants using The other classes in the `gui` directory.

Finally, the `main.py` file in the root of the project, creates an instance of SmartTVAppGUI an runs it in an infinite loop.

8 Configuration

For the correct operation of the Raspberry Pi you will need to connect all the peripherals to be used, such as keyboard, mouse, monitor and then energize the Raspberry with a USB to micro B cable.

8.1 Operating System

The operating system recommended for this project is the 64-bit Raspberry Pi OS (Legacy) Lite (bullseye). Also, the project requires a list of libraries to be installed. This can be done by executing the **requirements.sh** script.

8.2 Requirements

- Tkinter
- pyudev
- Pillow
- libwidevinecdm0
- VLC
- xserver-xorg
- lightdm

After installing the requirements we have to change the raspi configuration to enable the booting process with the display manager. This can be done in the `raspi-config` boot section.

Then, reboot the system and run the `main.py` program to start the application.

9 Video evidence

Video evidence of the MaFE TV system functionality: **MaFE TV**

10 Conclusion

This project led us to learn how to set up and program a raspberry pi to use as a multimedia center and helped us develop some of the concepts and practices learned in class. The media center is still very simple, so it could be improved by optimizing the code, improving the graphical interface displayed to the user, adding support for remote control, and implementing new functionalities.

References

[1] *Sistemas Embebidos (Integrados): principales aplicaciones*. URL: `https://www.tecnologias-informacion.com/sistemasembebidos.html`.