Project Proposal - Unix

420-321-VA UNIX

Section 00001

Semester 03

Lydia Ayala Hernandez (2264230)

Aya Kharchafi (2368616)

November 10, 2024

1- Project Description/Goals

The project will consist of creating a camera using the Raspberry Pi 4 Model B and Camera Module 2. Our Raspberry Pi will be able to take pictures and videos whenever a person clicks a certain button. The pictures will be saved as a file and will be available on a web server. Using the command line, one will be able to add filters or crop their pictures and videos. There will be 3 main buttons: one button to take pictures, record videos, and one to power on and off the Raspberry camera. Our main goal was to create a portable camera, meant for hikers or bikers that would like to take pictures while doing their daily activity. Obviously, anyone else can use it as a sort of GoPro, so our project targets anyone who would like to record or take pictures without the need to hold the camera. The completed camera will be inside a case that we will create and customize to fit all the components and the buttons. The camera will be installed on a person's body using a harness and will be stable enough to not fall or move (drastically) when it's on someone's body. The harness will be resizable and comfortable so anyone can wear it for hours without feeling it. We will try to provide a good amount of memory so people can record for a couple of hours without the memory being finished.

2- Platform of choice: The project is built on the Raspberry Pi 4 platform, which runs on open-source operating systems like Raspberry Pi OS, derived from Linux. This platform offers a compact form factor, sufficient processing power for the camera Module 2 and web server operations, and flexibility for running custom scripts and managing peripherals, making it an ideal choice for this project.

3- Demonstration plan: Our project will be demonstrated using the complete portable camera system, built with the Raspberry Pi 4 Model B and Camera Module 2. The functionality of the tactile buttons for picture-taking, video recording, and powering on/off will be showcased. The captured pictures and videos will be accessible and displayed on a web server hosted on the Raspberry Pi.

4- Requirements:

Basic System Setup and Security

PART I - Setting up the Raspberry Pi

Hardware Components

- Raspberry Pi 4: Acts as the core computing unit.
- Camera Module 2: Captures images and videos.
- Storage Device (Micro SD Card): For storing OS and captured media files.

- HDMI Cable: For connecting to a display during setup and testing.
- Power Supply Cable: Provides reliable power to the Raspberry Pi.
- Tactile Buttons (3): Used for picture capture, video recording, and power on/off.
- Jumper Wires and Breadboard: To connect tactile buttons to GPIO pins.
- Portable Power Source: Ensures the camera's portability.
- 3D Printed Custom Case: Protects the components and provides a user-friendly form factor.
- Connectivity Module: For network access and uploading images.
- Cooling (Heat sinks/Fan): Helps maintain optimal operating temperatures.
- Accessories (Straps): Secures the camera to the user's body for stable and comfortable wear during activities.

Software Requirements

- Camera Drivers.
- Python libraries for GPIO and camera controls (picamera for camera control).
- Lightweight web server for media display.
- Gallery Software.

Setup Steps

1. Prepare the Raspberry Pi:
   Install Raspberry Pi OS on the micro SD Card and configure the basic system settings.

2. Install the Camera Software:
   Load necessary drivers and libraries for the Camera Module 2.

3. Attach and Test the Camera Module 2:
   Connect the camera module, then test its functionality by capturing taking images and videos.

4. Connect Tactile Buttons:
   Wire the tactile buttons on the Raspberry Pi's GPIO pins via jumper wires and a breadboard.

5. Write Python Scripts for Camera Functionality:
   - Picture Button: Captures still images.
   - Video Button: Starts/Stops video recording.
   - Power On/Off Button: Powers the Raspberry Pi camera on or off.

6. Powering the Camera with a Portable Source:

Connect a portable power source for standalone operation.

7. Choose and install webserver and gallery software.

8. Test and debug:
   Test button functionality and troubleshoot any hardware/software issues.

9. Build 3D Printed Case:
   Design and build a case that houses all components and ensures accessibility.

10. Add Straps for Stability:
    Attach a secure, resizable harness for comfortable and stable usage.

11. Add the optional Enhancements: heat sinks, to improve cooling efficiency, and ToF(Time of Flight) sensor, to calculate the distance between the camera and an object.

PART II - Setting up the Web Server

1. Host a Local Web Server:
   Configure a lightweight web server on the Raspberry Pi, for example: using Nginx.

2. Upload Images to an External Web Server:
   Add functionality to upload captured images to an external server for remote access.

3. Create a Web Interface:
   Develop/Install a simple interface to manage and display captured images.

Authentication method: Configure RSA key-based authentication for user access to the raspberry pi.
File permissions: Use chmod to control access of the pictures and video in the system.
Firewall configuration: Use ufw to set up basic firewall rules, allowing only necessary ports (for web server access) and blocking others to ensure system security.

<u>Process or Service management/scheduling: Systemd services</u>

- We will create a custom SystemD service file that launches necessary scripts or processes during the startup of the Raspberry pi, so the camera is always ready after boot.

Automated Tasks using a script language: Python

-Buttons
An automated task that continuously monitors a script in Python to detect button presses via GPIO pins, triggering actions like capturing photos, starting or stopping video recording, and powering on/off the Raspberry Pi.

-Count of pictures and videos taken
An automated task that counts and updates how many pictures and videos were taken with the Raspberry Pi portable camera, we will use a script in Python running as a daemon.

5- Major Technical Solutions Compared

Storage:

1-SD card for local storage: An SD card is a convenient method to store media. A card can vary in price depending on the amount of storage and the brand, but mostly it provides a good amount of storage for a reasonable price. The downside of this type of storage is the probability of getting electronically corrupted and the metal part of it is very sensitive, which can lead to it being broken, which then becomes unusable. Of course, the chances of it getting corrupted are pretty low.

2- Cloud Storage:
Cloud Storage will be a great option for storing all the media files that will be taken with the camera system, but since we don’t think we will be taking more than like 300 pictures and videos, we won't need cloud storage, since it is recommended for big storage for systems. If the product becomes popular and big, using cloud storage would be ideal.

Additional Components:

1- Heat Sink: The Raspberry Pi 4 does not require a cooling system, but if we want to max out its capacities or use it for longer periods, installing a single heatsink could be a solution to cool the Raspberry Pi. A heat sink is better than a fan since absorbing the heat is more efficient than blowing the heat away. The only downside would be that dust could accumulate, which a fan could fix but as I mentioned, a cooling system is optional.

2- Time of Flight sensor: If we have time, adding a Time of Flight sensor would be great to be able to calculate the distance between the camera and a certain object. We would program it to be able to tell the person if they are too close to an object and if they take a picture, the image can be blurry or not visible. This option needs to include some sort of alert to indicate the closeness to the object, so it is an option that we can consider if we finish sooner than expected.

Remote Access:

1- Private Webserver:

- Apache: A highly customizable and configurable. It is reputable for its traditional LAMP stack setups
- Nginx: Very lightweight and fast, and ot handles high-concurency traffic better than apache, however we are making a public webserver where anyone can enter, so this feature is not something that we should worry about. It's small memory footprint makes it highly compatible with the raspberry pi. We won't need much resources and we won't be publishing a thousand pictures or images so a lightweight server fits our project

2-Gallery software:

- Lychee: Access photos from the browser which is easy to use. Doesn't require a database server. If we want to use a mobile to view the images and the environment, it provides a mobile-friendly design. It organizes photos in albums to keep everything clean and organized. It basically is a photo manager, so picking this option would facilitate the management of our media.
- PiGallery2: This software was designed for Raspberry Pi, so it fits more our project than the other gallery softwares. It is lightweight and provides the client with so many functions and tools: Search bar and filters to search for specific images. It is mobile and desktop friendly. We have learned about docker, and this software can be edployed using it. It creates or generates thumbnail for images so it makes our job easier and creates a clean environment. Finally, if we want to pair it up with a webserver, it is compatible with Nginx, and it could work with apache, but it works best with static web servers. It needs Node.js for it's backend. This is an extra installation but it is can be simple.
- PhotoPrism: More advanced than the previous options. It manages pictures using AI to search or filter images. The AI can be used to search images depending on faces, objects and more. It works well for mobile or desktop use. A docker would be recommended, if paired up with Apache and Nginx.

Media Customization

1- ImageMagick: We will be making filters available through the command line. ImageMagick is a free, open-source software used to edit images from any range of file formats (JPEG, PNG, TIFF, etc). They make editing available through the command line which is our goal. ImageMagick is also one of the most popular image editors that works using the command line. People compare this software to Adobe Illustrator, which proves that it is a powerful editor that we can depend on.

2-OpenCV: OpenCV is open source and it is used for Object detection and Video processing. It is powered by AI and supports Machine Learning. This can be used on the command line on a Raspberry Pi. It will be a great option if we use face recognition while recording. We can also implement this software and create several folders for the different objects that it detects and publish them on the web server.

Power Source

1- Battery Pack: We can add a portable battery pack since our main goal would be to make our Raspberry Pi portable. We are leaning towards a battery pack since it is one of the most common options when creating a power source. We would need to find the required amount of power and capacity to keep our Raspberry Pi running for longer periods.

2- Solar Power Controller: Providing power to our battery with a solar charger would be a great eco-friendly option, however, it is an extra component that is not necessary and would cost more. Also, it will need to be compatible with the battery and should provide enough volts for the Raspberry Pi. This extra component can also add weight to the camera, so it would be less convenient for a camera that is attached to someone, but it is meant to be outside so the solar charger is a good idea.

6- Timeline

Week #1 (Sunday 10 - Friday 15) :
- Connect the camera to the Raspberry Pi
- Install the tactile buttons and complete the script
- Start the web server

Week #2 (Sunday 17 - Friday 22)
- Finish server
- Debug and test
- Make the camera portable
- Send media files to the web server
- Start creating the case for the completed camera (3D printing)

Week #3 (Sunday 25 - Monday 2)
- Finish the case
- Attach the camera to the harness
- Prepare and finish the content for our presentation
- Go over everything to make sure that everything works

7- Team Composition

- Aya Kharchafi, 2368616
- Lydia Ayala Hernandez, 2264230