Shaheryar Anwar

Journal 3

November 26th: Automating the Script and Managing Storage

Today, I aimed to automate the Python script so that it would run automatically on boot. I learned about systemd and created a service file for the script. This process was tricky because I had to ensure that the correct paths for both the Python interpreter and the script were specified. After some work, I successfully set up the script to run on boot, which was an important step toward making the system autonomous.

I also wanted to manage the stored images to prevent the storage from filling up. To do this, I implemented a cron job that deletes images older than 7 days. It took some time to understand the find and rm commands to identify and remove files, but once configured, it worked well to keep the system's storage in check.

Reflections:

- Automating the script with systemd was a great learning experience. It helped me understand how to manage background services in Linux.
- Implementing cron jobs to handle image deletion was a practical solution for managing storage space.

November 27th: Testing and Real-World Adjustments

I tested the system in a real-world setup by placing the Raspberry Pi and PIR sensor on a table to monitor motion. While the core functionality worked, I noticed that the sensor was still triggering false positives, likely due to background movements or temperature changes. To resolve this, I adjusted the sensor's sensitivity using its potentiometer and experimented with its placement. This helped minimize false triggers but didn't fully eliminate the issue. So, I had to replace the sensor with another PIR motion sensor, which fixed the problem.

In terms of managing images, I monitored the cron job, which successfully deleted older files as expected. However, I realized that adding a notification system to alert me in case of system failures or issues would be beneficial in the future.

Reflections:

- Real-world testing revealed how environmental factors affect PIR sensor accuracy, and I had to experiment with placement and sensitivity.
- The cron job for image deletion worked as expected, but I considered adding monitoring features for future improvements.

November 29th: Final Adjustments and Presentation Preparation

I finalized the system for presentation. I tested it one last time to ensure it could run reliably for extended periods. I also made some adjustments to the sensor's placement and sensitivity to ensure it detected motion more accurately.

For the final demonstration, I prepared a presentation explaining the project's functionality, the challenges faced, and the solutions implemented. I also created a user manual outlining how to set up and use the system. Although the system worked well, I thought about adding additional features, such as a notification system for real-time updates in the future.

Reflections:

- Preparing for the final demonstration helped me reflect on how the project evolved. It’s not just about coding—it’s about making sure the system works in real-world conditions.
- Testing and fine-tuning the sensor was crucial for accuracy, and I learned a lot about the importance of hands-on adjustments.

Conclusion:

This project taught me a lot about hardware interfacing, automation, and troubleshooting. Through hands-on work and testing, I gained practical skills in Python programming and system administration. I’m pleased with how the project turned out, and I’m looking forward to future improvements, such as adding a notification system to monitor the project remotely and fixing the random motion detection issue.

Error:
As of **November 30th**, the PIR motion sensor in my Raspberry Pi, which had been working before, stopped functioning and started causing the same problem I encountered on November 27th—false movement detection. I tried adjusting the sensitivity of the PIR sensor, but it still didn't work. Since this was my last PIR sensor, I’ve done my best to fix the issue, but I have had no luck.

As of **December 1st**, when I tried again in the morning, the same issue persisted, meaning it still wasn’t fixed. I asked my partner whether his sensor was working, but he also reported that his sensor wasn’t functioning properly.

Possible Causes:
I searched online for possible reasons why the PIR motion sensor might be malfunctioning, and I found some potential causes:

1. **Electrical noise:**
   The sensor may be picking up electrical noise from nearby devices or from the Raspberry Pi itself, which could falsely trigger motion detection.
2. **Ambient conditions:**
   Temperature, light, or heat could be interpreted as motion, causing false triggers.

Shaheryar Anwar

Summary:
The issue is likely caused by environmental sensitivity or electrical noise, which can cause the PIR motion sensor to detect movement even when there isn't any.

Reflection on Meeting Original Expectations

When we started the motion detection project using the Raspberry Pi, our original expectations were to create a functional system that would detect motion and respond by capturing images. We outlined a clear set of deliverables: setting up the Raspberry Pi, integrating a motion sensor, and capturing images when motion is detected.

Meeting the Proposed Plan

Our project successfully met several key expectations:

- Hardware Setup: The Raspberry Pi, motion sensor, and related components were assembled and connected.
- Software Development: We implemented scripts to monitor motion and trigger the camera module to take pictures, as planned.
- Collaboration: Working with Amir ensured that tasks were distributed effectively, enabling smooth progress throughout the project.

Divergences from the Original Proposal

Some aspects of the project diverged from our initial proposal:

1. Scope Adjustments:
   Initially, we considered integrating advanced features like storing images in a cloud database or adding real-time notifications. However, due to time constraints, we focused on local storage for image files.
2. Hardware Additions:
   We purchased an additional USB-C to USB-C cable, which was not part of the initial plan but became necessary for streamlined testing and system redundancy.
3. Software Tweaks:
   We had anticipated using pre-built libraries for motion detection, but challenges with library compatibility led us to write custom Python code for sensor integration.

Reflection on Changes and Outcomes

These adjustments allowed us to deliver a functioning motion detection system while staying within our time and resource limits. While we did not fully implement all the advanced features, and the system didn't work perfectly, the foundational system we built is scalable for future enhancements.