IOT Term Project Report

Project:

Motion Sensor Cat Gate

Team Member:

Chaoyi Wang

Problem Description

I have a cat in the house. Although he can walk anywhere without my permissions, there are certain areas in the house that I do not want him to pass. For example, if I find there are ants crawling on the floor, I want to spay ant killers to kill the ants. But in the meantime, I also want to make sure that my cat won't have access to the areas where ant killers just applied. I am not willing him to accidentally lick or sniff the ant killers. Traditionally, I would use many pieces of pet gates to enclose the restricted area. But that's not a perfect solution, pet gates are expensive and I need more than 5 pieces to enclose the area if it's the center of say places like the living room. The physical pet gates also occupy a large area of the storage. On the other hand, my cat can easily jump over the 3 feet high pet gates. So I need a budget friendly and space efficiency idea to stop him from entering the restricted area.

Solution & Deliverables

I came up with the idea to build an IOT device to virtually stop my cat from entering the area where he is not supposed to be in. This is a Raspberry Pi based project. The IOT device features two PIR motion sensor, connected to a speaker with an HDMI cable. Two PIR motion sensors can help differentiate whether it's a human or a cat passing through. Whenever a cat is detected, the device will trigger the alarm and make an annoying sound to scare my cat away. It connects to home wifi and can upload the detected data into a private cloud database. The database is set up on a NAS (Network Attached System) through the intranet. The IOT device will also send a weekly report to me through email. So I can view and track each recorded movement.

Requirements

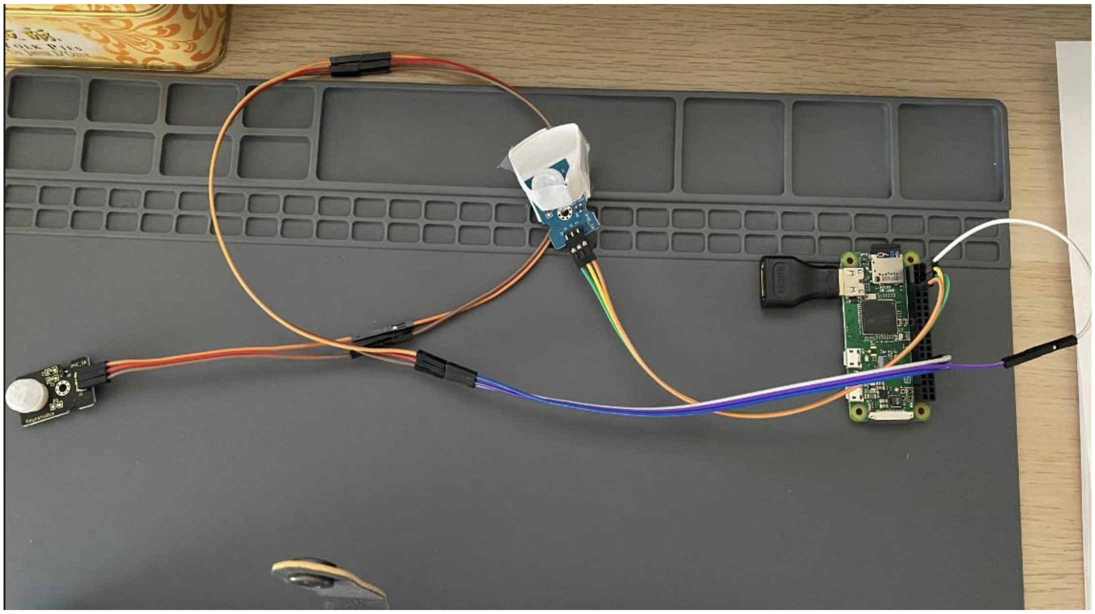
- The project requires the system to react based on events triggered by two motion sensors in a short time frame. So it requires callback functions to let one event wait on the other one.
- For database connection, python odbc is required for connecting python codes with mysql
- A database is required to store motion detection related data, such as who triggered the sensor, and the time the sensor was being triggered.
- Choose local database over cloud database because of security concerns and project budget. I set a static IP address for the Raspberry Pi Zero used for this project, only open mysql's port 22 within the intranet and set the remote access to the database for the RPi Zero's IP address only.
- Used an HTML template and Python's Beautifulsoup module to generate the email content.

Related work

- I tried to divide and conquer each individual problem or difficulty.
- Since I used the lite version of Raspbian, I researched on how to config the Wifi connection, enable RPi features like SSH tunnel through the command line.
- I also researched on how to get signals from two pins over time.
- The original audio connection I made was connect the Raspberry Pi with a speaker through bluetooth, but the connection is sketchy and unstable, so I turned away to seek for ways to do wired audio connection, after some research, I found out that it is harder to connect a speaker through the GPIO Pins and easier if just connect the speaker through the headphone jack, but a Raspberry Pi Zero does not have a headphone jack, I researched on alternative ways to solve the problem.

Architecture

- Two PIR Motion Sensors: sensor 1 S Pin connected with GPIO Pin 12
- Sensor 2 S Pin connected with GPIO Pin 4
- A speaker connected to the micro HDMI port via a 3.5mm analog to HDMI adapter

- 

Software/Hardware Components Used for the Development

- Hardware:
    - Raspberry Pi 4B (for testing)
    - Raspberry Pi Zero W (for production)
    - PIR Motion Sensor x2
    - GPIO Female to Male Connector
    - Soldering Machine
    - A Working Computer (Remote Control of Raspberry Pi Zero)
- SoftWare/Packages
    - Python:
        - Mysql.connector, datetime, RPi.GPIO, email.mime, beautifulsoup, smtplib, ssl, threading, schedule
    - JavaScript / HTML / CSS

Work

- This project was solely done by me

Conclusion

- **Limitation**:
    - The most challenging part is to get two motion sensors working properly, my current project setup is still sometimes buggy as I need to determine a better trigger frequency for the sensors and have the sensor work within the same time frame.
- **Future Work:**
    - The project currently still needs to be started manually, I will work on this and make the program a background process so it can start automatically on each time the Raspberry Pi boots up
    - Will build a front-end website for checking detection logs in real time
    - Make the object detection more accurate with different types of sensors.