Smart Fan

Gavin Li, Eric Chen, Mario Tafoya
Professor Zhou Li
Department of Electrical Engineering and Computer Science

Goal

Our goal is to save electricity used by AC in California by making the current fan model more useful.

Introduction

Currently in California, tons of electricity is being wasted on air conditioning every hour. Fans are a much better choice in saving energy due to less electricity consumption and cheaper production.

$$P_{(W)} = \frac{E_{(kWh)} \times 1,000}{T_{(hrs)}}$$

Fig 1. Formula for power used in Watts

We can improve the current fan model with some new features:

- turn itself towards the user
- horizontal orientation
- auto shut-off / turn-on
- fan speed increase / decrease depending on the range of the user or temperature

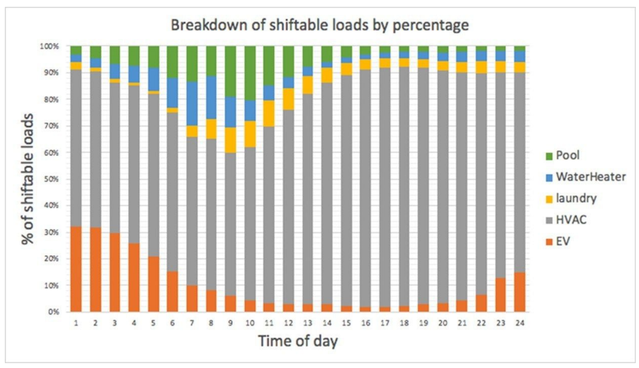

Fig 2. Electricity of an average home in California used by AC (HVAC)

Approach and Design

Our design is essentially the fan with the sensors stacked on top of it. Servos control the fan from the base, and the entire thing is connected through a breadboard to the Raspberry Pi and the software.

Hardware

The main piece of hardware will be the Raspberry Pi. This Raspberry Pi will need to be used in order to connect all our hardware components, and send signals from the Raspberry Pi software to the hardware so we can activate the components.

The camera is essential for tracking and will be attached to the rotation object so it will follow a moving object. It will be used with the OpenCV software.

The servos are necessary for rotation of all the parts. The camera / fan will be connected to a x-axis servo so it can move left and right.

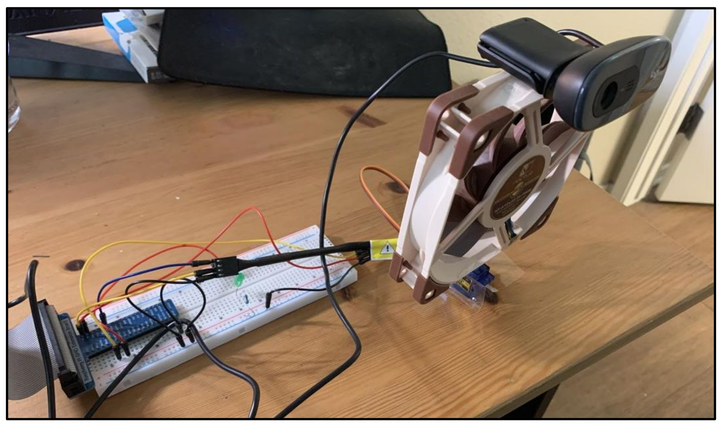

Fig 3. A prototype of the Smart Fan

Software

Python and OpenCV

We use Python with the OpenCV module as our software portion of the project. This module allows us to handle facial / body recognition so we can use the data gathered from the camera to send data back to the hardware to tell it where to move. It is the main skeleton of the project.

The program grabs video data, outputs it to the computer, and draws a box around the detected face.

```
# track the face while it is moving
def track_face(face_position):
    # servo to the left (our right)
    if face_position > scan_range_right:
        servo_left()

    # servo to the right (our left)
    if face_position < scan_range_left:
        servo_right()

    # jitter problems
    time.sleep(.05)
    servo.ChangeDutyCycle(0)
```

Fig 4. Sample of code used for face tracking

Acknowledgements and References

[1]Sense, “Analysis of energy data in 1100 California homes shows that residents can help prevent rolling blackouts by taking simple actions during critical evening hours,” www.prnewswire.com. https://www.prnewswire.com/news-releases/analysis-of-energy-data-in-1100-california-homes-shows-that-residents-can-help-prevent-rolling-blackouts-by-taking-simple-actions-during-critical-evening-hours-301121208.html (accessed Nov. 11, 2022).

[2]A. Wigglesworth and R. Vives, “As Earth warms, air conditioning use in California could exceed power supply in next decade,” phys.org. https://phys.org/news/2022-02-earth-air-conditioning-california-power.html (accessed Nov. 11, 2022).

[3]“INSTRUCTION MANUAL Standing Fan.” [Online]. Available: https://images.homedepot-static.com/catalog/pdfImages/58/58dc84c8-734d-4e42-9e71-5ecc392d14c4.pdf

The Henry Samueli School of Engineering
University of California · Irvine