IT'S LIT

CARA PEW AND ASHLEY DATTALO (CLASS OF 2017)

Report V1, 12.07

Cal Poly CSC 458 Project Presentation

Final Project - Proposal

- Motives: laziness + interest in home automation
  - Getting up to turn the light on or off can be a pain
  - We wanted to see if we could tackle this problem
- Solution: Lights automated by:
  - Motion
  - A web app
- 2 Raspberry Pis + 2 Ultrasonic Sensors + Philips Hue Bulb & Bridge

Functional Specification

Requirement	Feature Name	Detail Description
1. Ultrasonic Sensors	Support Detection of Movement	Detect a user's movement. Using two will allow for understanding of whether a user enters or leaves a room.
2. Raspberry Pi Server	Support Communication between Ultrasonic Sensors	This allows the ultrasonic sensors to send their data to one place for decision making.
3. Bridge	Allows Communication to Philips Hue Lightbulb	The bridge allows the services to connect to the lightbulb over the network
4. Smart Lightbulb	Response/Action	Acts based on commands given from other services
5. Web Server	Allows User to Control Lightbulb Directly	With a web server, a user can control the lightbulb in more custom ways.

Functional Specification

- Many similar projects out there, but we had never implemented something like this
  - Ultrasonic sensor examples from Raspberry Pi Tutorials, Server example and code from Flask
- Most smart lightbulbs not equipped with motion sensors

Technical Details: The Website

- On/Off
  - Makes a PUT call to the bridge
    - Ex: http://129.65.221.203/api/353ae7b31/lights/1/state
- Motion Sensing On/Off
  - Makes a PUT call to a route our own server
    - Accessible by Raspberry Pis
- Brightness
  - {on: true, bri: 254}

Technical Details: Raspberry Pis

- Both Pis have ultrasonic sensing code: calibration period, then sense when object is closer than usual
- Server on one Pi, both Pis send times sensed from ultrasonic sensors to server
- Server then checks whether motion sensing should happen or not (from our web server)
- If so, then send bridge call to turn bulb on or off based on which Pi’s time was first

Technical Details (continued)

- Challenges:
  - Had never programmed on a Raspberry Pi before
  - Had never programmed using sensors before
  - Cal Poly network issues
  - Understanding local area networks, how to connect devices to each other
  - Didn’t realize bridge was necessary
- Solutions:
  - LOTS of research
  - Help from professors
  - Experimenting
  - Amazon Prime

Technical Details (continued)

- Methods for Goal Completion:
  - Start small– divide project into parts
  - Changed project goals as our understanding of the hardware changed
  - Made sure everything worked before moving to next task
  - Experimented with lots of different technologies/methods
  - Found a lot of useful information online
  - Split work into parts by previous knowledge– Cara more on ultrasonic sensors, Ashley more on web server

Demo

Conclusion

- Future work:
  - Potentially work with colored lightbulb? Music?
  - Perfect ultrasonic sensing– more accurate
  - Host website to be accessible from any network
- Takeaways:
  - We learned a lot about REST, servers, APIs, IP addresses…
  - IoT lets you do cool stuff!