The Occupancy Project

Shaun VanWeelden, Lee Robinson, Mike Pruszinske, Sam Ennis

Iowa State University – Senior Design May '16 – Group 36 – Faculty Adviser: Mani Mina

WHERE'S AN EMPTY ROOM?

Every finals week, students have trouble finding places to get together to study exam materials and finish group assignments. Not only that, throughout the year, school administrators try to make informed decisions about room and power usage for on-campus buildings.

We focused on developing a solution to assist both students and school administrators with some intuitive software and hardware innovations.

WHAT WE NOTICED

This occupancy sensor is in almost every classroom at Iowa State to turn the lights on and off. Wouldn't it be cool if we could make it "smart" by gathering it's data to store, process, and visualize online?

OUR SOLUTION

Rooms around the university will report their current occupancy status via our occupancy sensor add-on to a user-friendly web application. With this new room occupancy information, students can make informed choices about where to study and meet, and the university can make informed decisions using our application's analytics about classroom usage, room popularity, energy consumption, etc.

Because of Iowa State's continued growth in enrollment, managing classroom usage effectively is more important than ever. The Occupancy Project will help Iowa State get the most out of their current buildings and be at the forefront of smart and sustainable technologies.

DESIGN REQUIREMENTS

Functional Requirements:

- Must work alongside existing occupancy sensors
- Hardware device should accurately report room occupancy status
- Application provides historical data about room occupancy

Non-functional Requirements:

- Added device components should cost less than $40 total
- Device should last at least one year without needing maintenance
- Software should display historical analysis and have an admin panel

Operating Environment:

- Our solution is ready to be implemented at Iowa State and is architected in a way to work with other institutions easily

DESIGN APPROACH

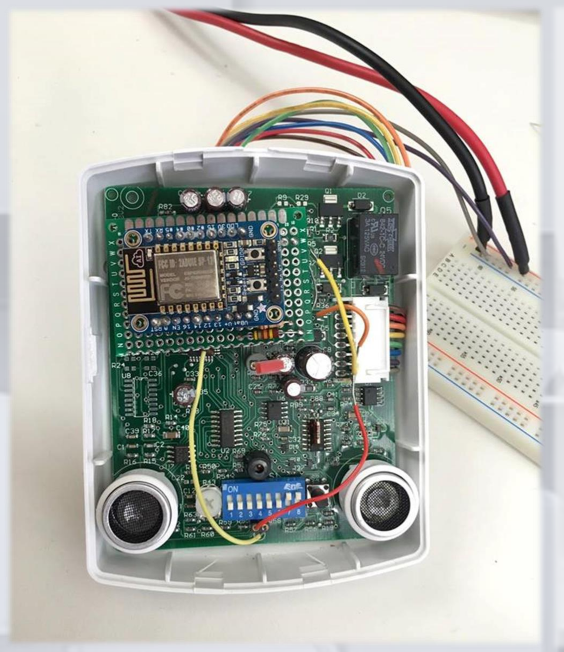
We initially brainstormed a solar-powered occupancy sensor add-on attached to the exterior of a sensor transmitting over Wi-Fi to our software system.

To the right is our first try.

We evaluated software and hardware components for cost, ease of integration, longevity, and electrical and spatial requirements.

As we built out our application, several initial ideas had to change before we decided on our final Hardware Implementation below

HARDWARE IMPLEMENTATION

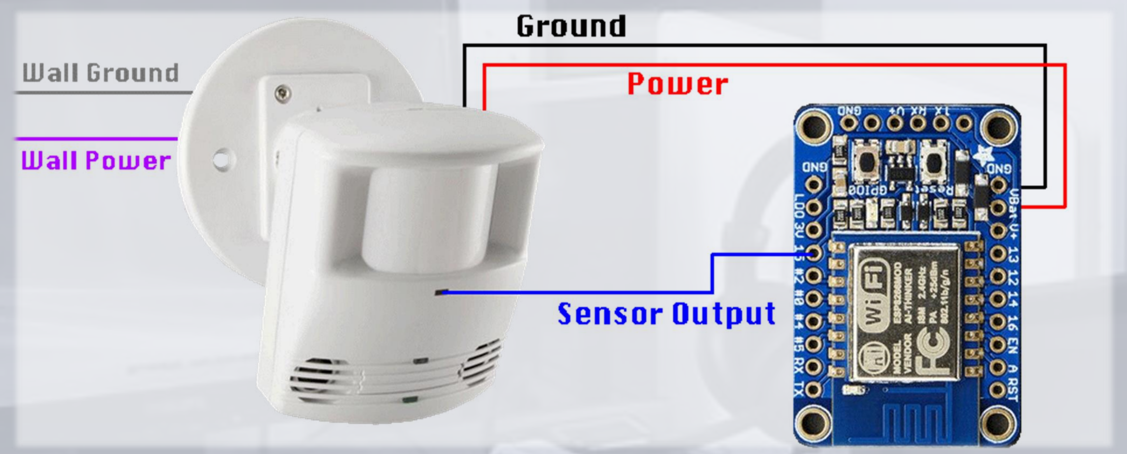


Our final hardware design involves the device integrating directly with an existing occupancy sensor.

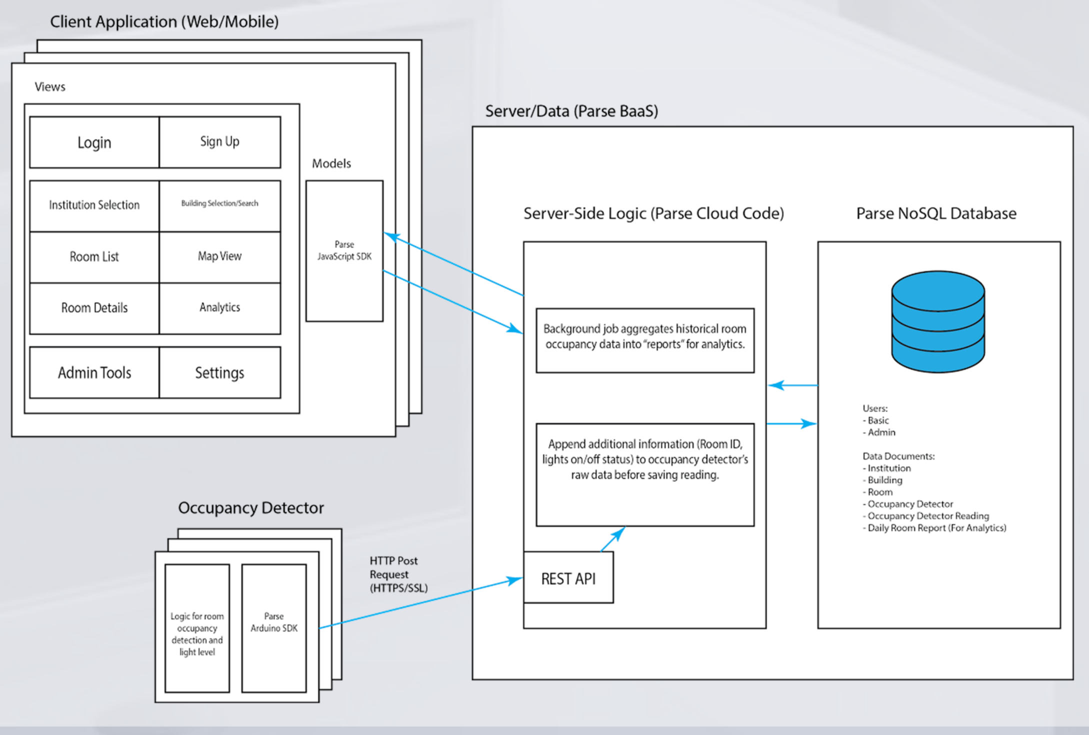
We use the voltage that powers one of the debugging LED lights to signal our microcontroller that motion has been detected.

Once our servers receive this, we log and visualize the data accordingly

HARDWARE WIRING OVERVIEW



SOFTWARE IMPLEMENTATION



TECHNICAL DETAILS

Programming Languages

- C – used for our embedded systems code
- JavaScript – used to interact with Parse Database
- HTML/CSS – used to visually structure our web application

Libraries

- Arduino – Allowed us to develop hardware component
- Parse – Hosted NoSQL database and sever logic
- jQuery – Reduced time needed to make site interactive
- HighCharts – Used to create visually appealing graphs

Development Tools

- Arduino IDE to assist writing embedded systems code
- Atom, Brackets, and Sublime were our favorite text editors for coding

HOW IT WORKS

Our ESP8266 microcontroller takes advantage of the Arduino library which allowed us to more easily program and configure the device. Using a REST API, we are able to communicate with a NoSQL backend service called Parse. After compiling this data in the backend, we use JavaScript to pull the data and display it on our website. Using HTML and CSS, we are able to display the data in list and graph formats, allowing the users to easily visualize which classrooms are available as well as review historical analytics data. To help populate our database, we created web scrapers to get classroom data from Iowa State websites.

INTENDED USERS & USES

We focused on two main users: Students and Administration

Students:

- Be able to quickly see room data including whether white boards, projectors, and more are in the classroom
- Filtering and maps help users find the best room for them

Administration:

- Be able to analyze historical data to make better administrative decisions for building planning
- Measure room efficiency with downloadable data

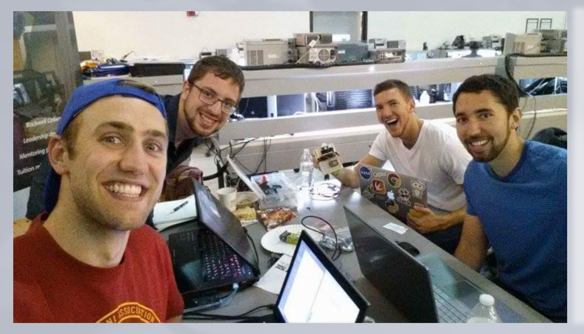
PROJECT RESOURCES

Each hardware device consists of:

1 x Watt Stopper DT-200 Occupancy Sensor ($30)
1 x ESP8266 Microcontroller ($10)
1 x 51 & 1 x 24 kOhm resistor ($0.20)

Total cost for each hardware device: $40.20

THE TEAM



Sam Ennis
Shaun VanWeelden
Lee Robinson
Mike Pruszinske

Email us at may1636@iastate.edu

See it in action

TESTING

Phase I – Preliminary Functionality

Goals:

- Ensure network connectivity from device
- Determine battery life of device using 9V battery
- Confirm occupancy status updates appear on website

Methods:

- Monitor device output & text log to determine connectivity
- Device pings website until it dies, check last ping time
- Visit website and verify changes are shown visually

Results:

✓ Connectivity successful. Repeated 10+ times
X Battery life was 42-48 hours when constantly pinging
✓ Web application successfully shows updated status 25 times

Phase II – Final Functionality

Goals:

- Determine solar panel + capacitor voltage/current output
- Test device when connected directly to host device power
- Test continued network connectivity
- Test scalability when adding many sensors & classroom data

Methods:

- Use EE equipment to test hardware capabilities and output
- Wire device to building power from existing sensors
- Monitor logs to determine connectivity status
- Spoof multiple sensors, data, and simultaneous pings

Results:

X Solar panel performance abysmal. 40-50mV and <.200mA in classroom conditions.
✓ Device successfully powered and operated
✓ Connectivity successful. Repeated 10+ times
✓ Web application successfully continued to work

Phase III – Anomalies & Security

Goals:

- Test what happens if wifi/network goes down temporarily
- Analyze security/public accessibility of device
- Test access to database restricted to approved devices

Methods:

- Monitor device output logs after disabling wifi and power
- Manually turn off device and observe how database/web app handles situation
- Attempt to send spoofed/fake pings to our database

Results:

✓ Device successfully reconnects to network and continues reporting
✓ Device maintains a sufficient level of security, unable to be accessed
✓ Database maintains a sufficient level of security, unable to gain access