Group 36: **The Occupancy Project**

Advisor: Mani Mina, Client: Mani Mina
Team Email: may1636@iastate.edu

Group Members:

Shaun VanWeelden - *shaun314@iastate.edu*
Sam Ennis - *sennis@iastate.edu*
Lee Robinson - *leerob@iastate.edu*
Mike Pruszinske - *mikep@iastate.edu*

Table of Contents

Project Statement

Recently, homes and offices around the country are rapidly becoming connected to the web via the Internet of Things, providing people with previously unknown information that can offer a wide variety of benefits. We want to provide these benefits to Iowa State University, as well, by making its lecture halls, classrooms, and offices “smart.” Our project makes use of IoT data to reveal classroom availability information to both Iowa State students and staff.

Rooms around the university will report their current occupancy status via an occupancy sensor 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.

System Design

Our entire ecosystem has three main components: the existing hardware, the device, and the web portal. Successful communication between these three components is crucial. Our device essentially acts as a bolt-on upgrade for the existing hardware. This allows our device to be agnostic to any existing sensors. As long as the sensor has a status LED reflecting a change in movement, our bolt-on solution will work. As referenced below in more detail, our device then communicates with the web portal allowing users easy access to data we’ve collected. This entire ecosystem was designed to be cost-effective and easily scalable.

Minimum System Requirements

- (1) Pre-existing motion sensor hardware
- (1) Our “Occupancy Sensor” hardware device
- IASTATE Wi-Fi network
- User

Input Specification

The input to our occupancy sensor is the photoresistor reading. This analog value ranges between 0-100, with the value increasing with the intensity of the light. When this value exceeds our trigger threshold, it updates the database with the new state.

Output Specification

The output of our occupancy sensor is readings sent to the database. Our sensor communicates via wifi with our NoSQL database (Parse.com). When the input value passes the threshold and is triggered, we use the Parse REST API to send the updated state to the database. This value is either “true” or “false”. Each device has a MAC address that allows the database to know which boolean value to update.

User Interface Specification

The user interacts with our application via either a web application or a mobile application. Both the web application and mobile application have identical functionality, but are written with separate code bases in order to enhance a user’s experience for the specific device it is accessed from.The application displays floor maps of buildings that we have wired up with our occupancy sensors. The building map will display the occupancy status of these “smart” rooms, pulled from our database, with either a green dot for “room open” (not occupied) or a red “x” for “room occupied.” The user is able to click on the room to view more details about it - its capacity, amenities it contains, etc., as well as view the room’s reservation status and historical usage (analytics).

Hardware Specification

Our device (ESP8266 Microcontroller) is powered by a 5V battery. This battery allows the device to connect to the wifi network. This powered state is toggled by a simple switch that allows us to conserve energy when not in use. A future improvement of this will allow power to be interrupt driven to put the device in sleep mode when not receiving new input.

Software Specification (Arduino)

The software, specifically the Arduino, has two main states: setup and loop. Our setup involves establishing a Serial connection for debugging. We communicate that we are setting up the device and connecting to the network. Next, we attempt to establish a connection to the IASTATE wi-fi network. If that connection was successful, we send out our initial state, which is false. The next state is the loop phase. This section repeats in the background for the entire lifecycle. Our software is continuously monitoring the input specification, and then in turn potentially producing the output specification.

Simulation/Testing

- Initially sent placeholder data to Parse to test network connectivity
- Created system to ping database from website to simulate data for new rooms
- Attached sensor to real hardware in Coover to demonstrate application
- Simulated a variety of devices throughout Coover for site functionality presentation
- Tested updating state in database reflecting change in display on website

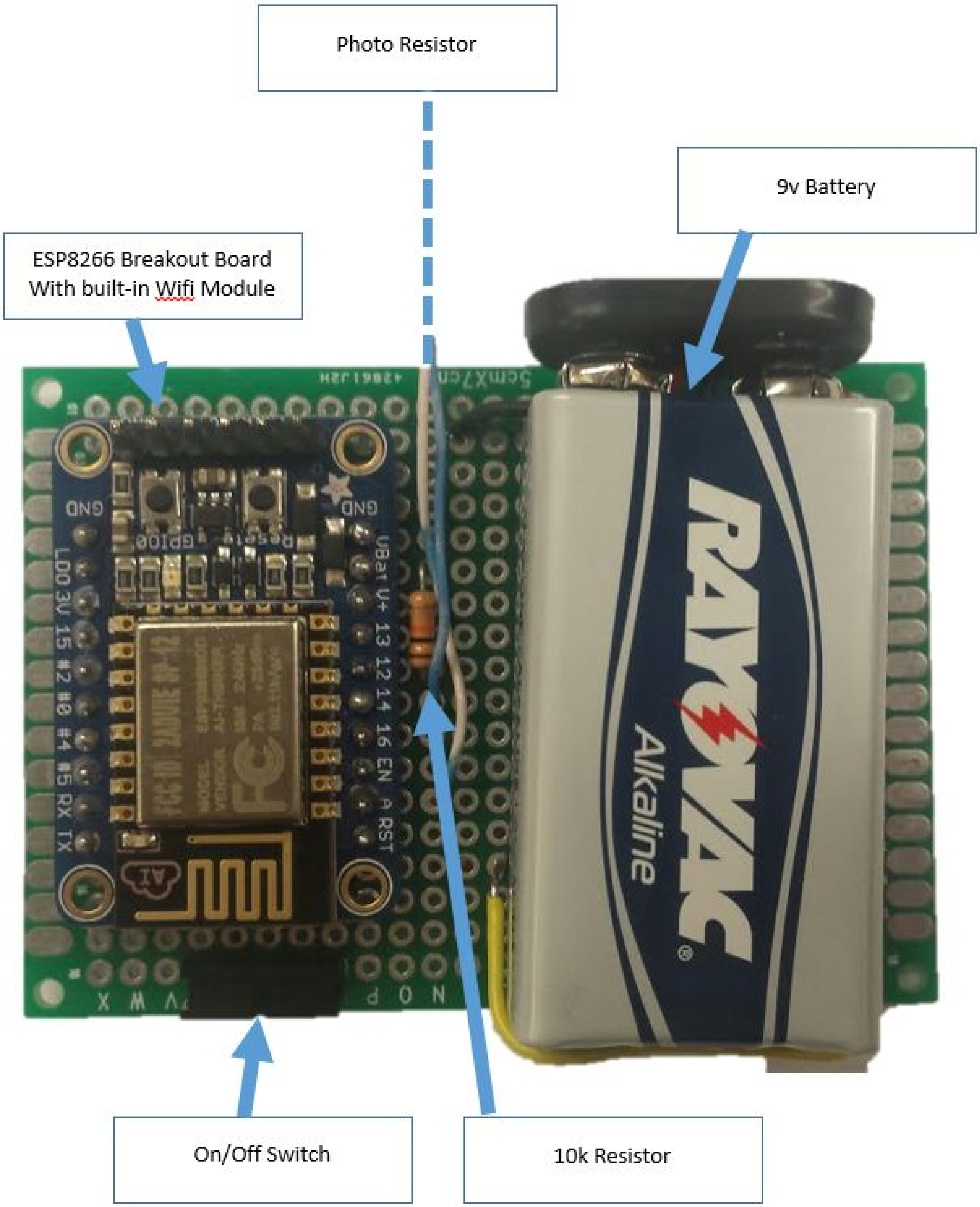
Prototyping

We have created a prototype of our hardware device that will closely mimic the real thing. The prototype of our physical device currently latches onto existing motion detectors and can detect the LED that flashes every time the existing detector detects motion. When the LED flash is detected, it sends the data to our database via a custom API call.

On the software side, we have already set up our database to not only accept the data coming from our physical device the correct way, we have filled the database with supplementary dummy data so we can work on our software application.

The application currently supports adding multiple institutions, buildings, and rooms to monitor. Right now, we have all the data for Coover Hall to show what rooms are occupied programmatically.

Electronic CAD



Software Design Diagram