Position Tracking Using WiFi

Tyler Bayne, Nathan Conrad, and Mitchell Woolley

Advisor: Dr. Ralph Tanner

Sponsor: Dr. John Kapenga

Contents

Need Statement

- A system is needed to track workplace items and employees, as well as report emergencies inside a building. It is necessary for the system to be able to determine the exact room each item or employee is located in.

Scope

- Physical Device
  - Microcontroller
  - User Interface
- Central Server
  - Display a map with device locations

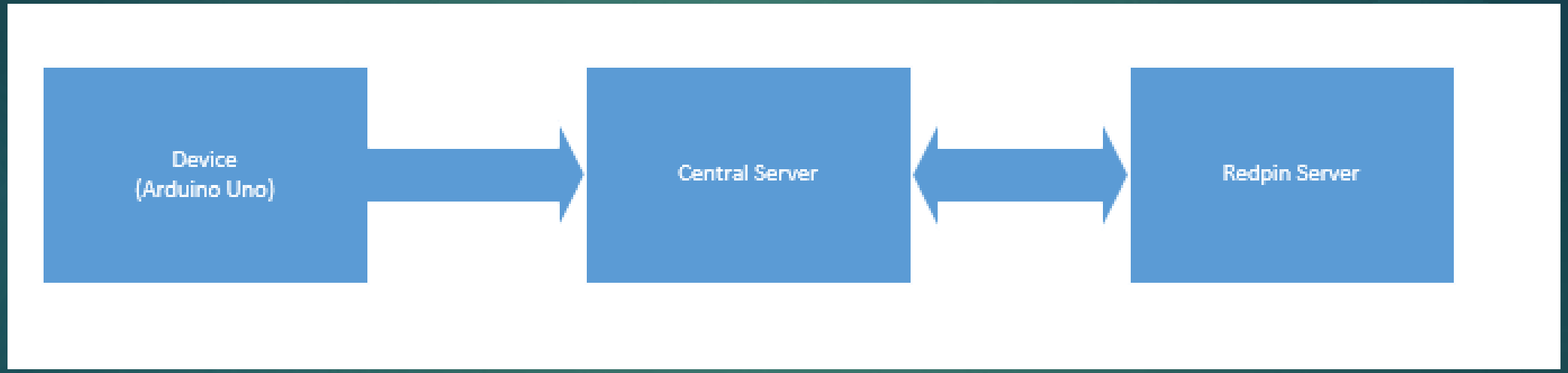
Specifications

- Physical Characteristics
  - The device should be relatively small
  - A final version will be attached to an identification badge worn by employees
- Functionality
  - The device should consume a low amount of power
  - The device should be in a low power state when it is not reporting its location
- Portability
  - The device should be portable
  - The device should be able to operate for an 8 hour workday

Specifications (cont.)

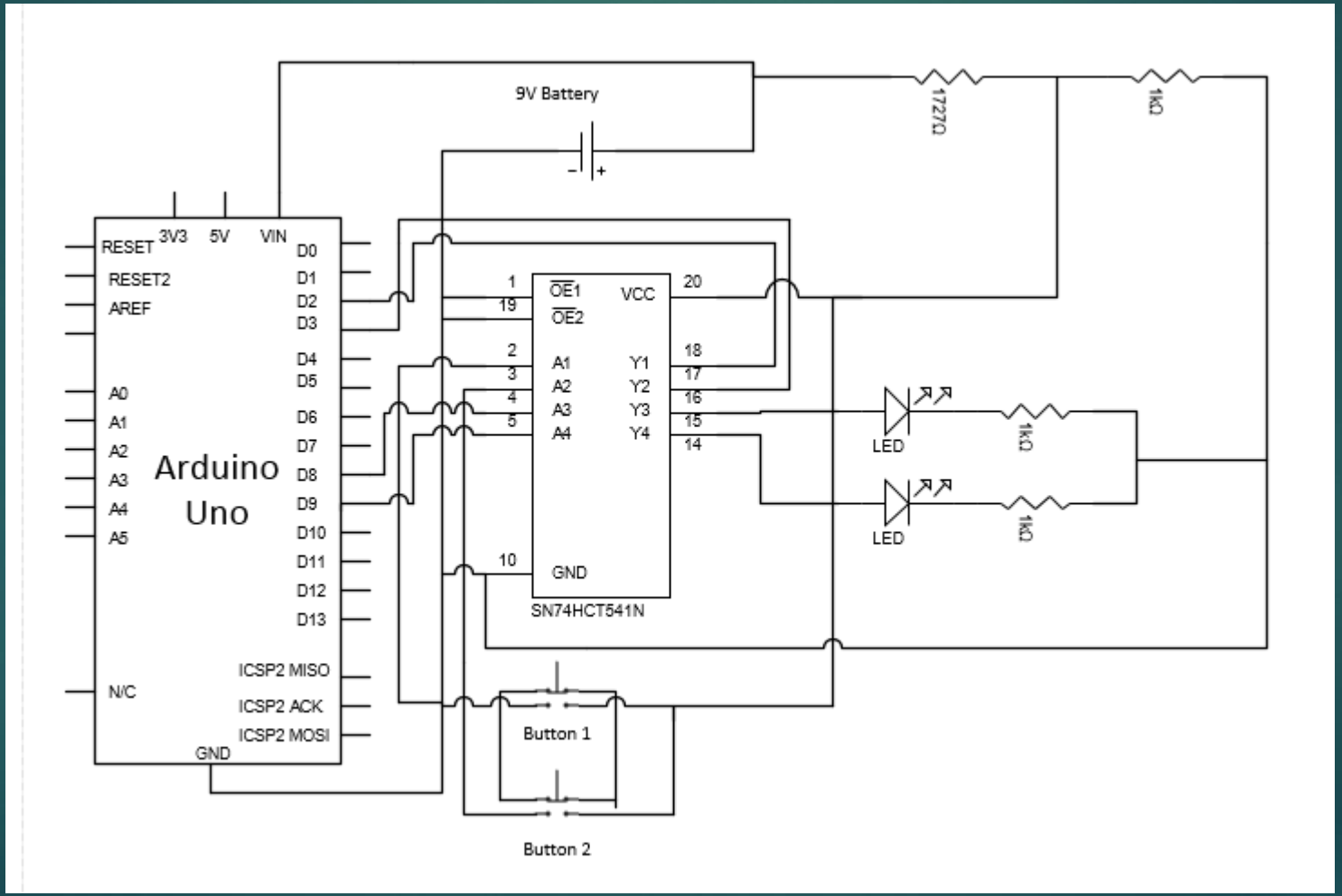
- Communication
  - The device should be able to communicate on the 802.11 g and g bands
- User Interface
  - The device should allow users to send simple messages to the Central Server about an emergency and/or their location
- Position Tracking
  - The device should periodically report its position to the Central Server
  - The location of each device should be shown on a map of the building
- Cost
  - The device should be inexpensive

Design Concept



Design Concept (cont.)

- Schematic

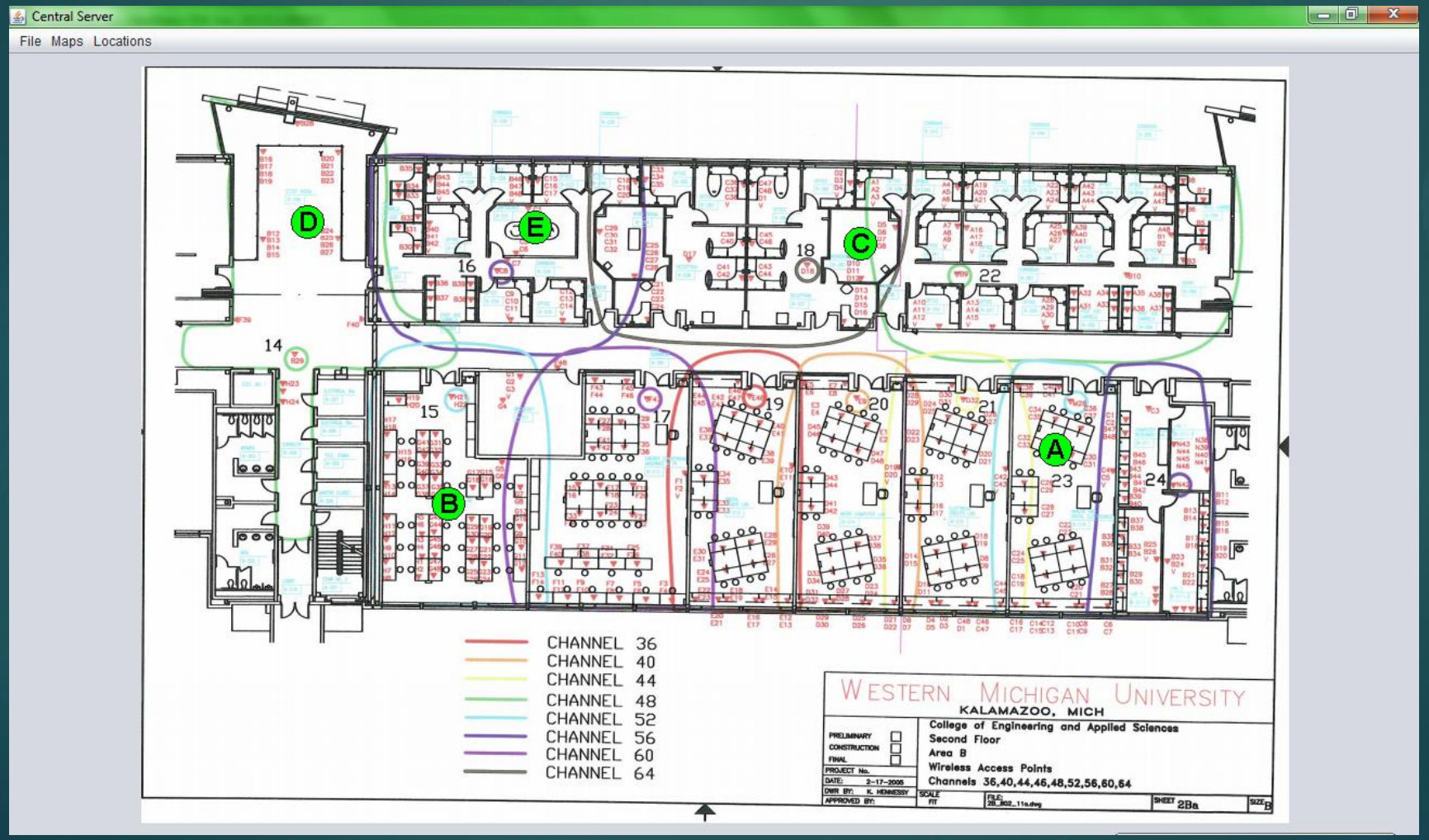


Design Concept (cont.)

- Redpin
  - Open source
  - Initial calibration phase
    - Receives numerous measurements of signal strengths at different locations
    - Creates “fingerprint” for each location
  - Calculating location
    - Receives new signal strength measurement
    - Uses “fingerprints” to calculate current location

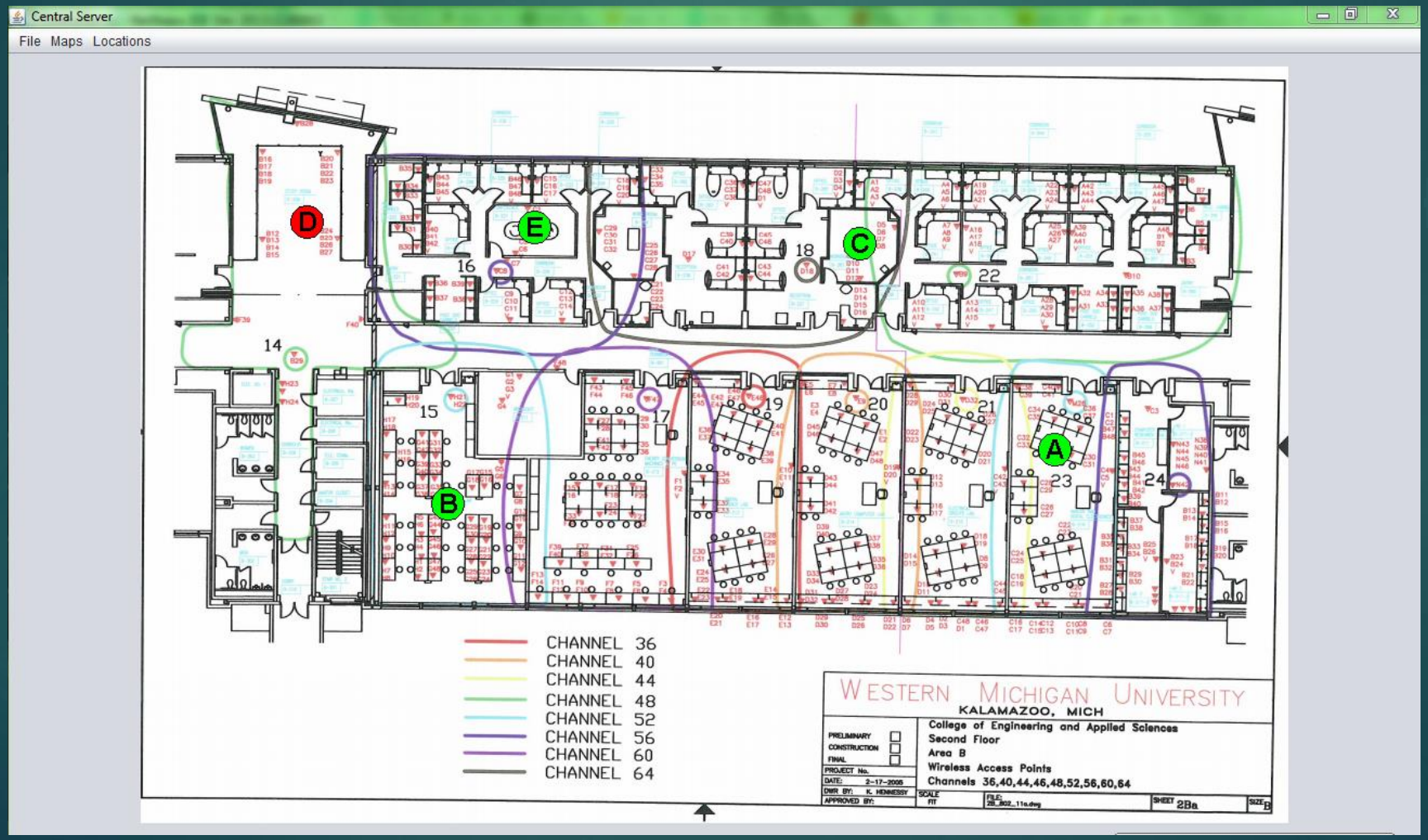
Design Concept (cont.)

- All devices are green
  - No emergencies

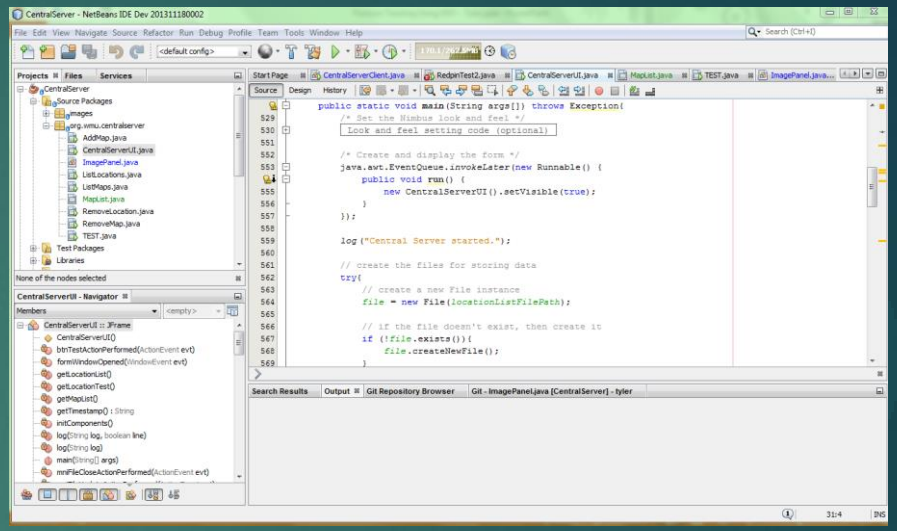


Design Concept (cont.)

- D has an emergency
  - Changes to red



Methods

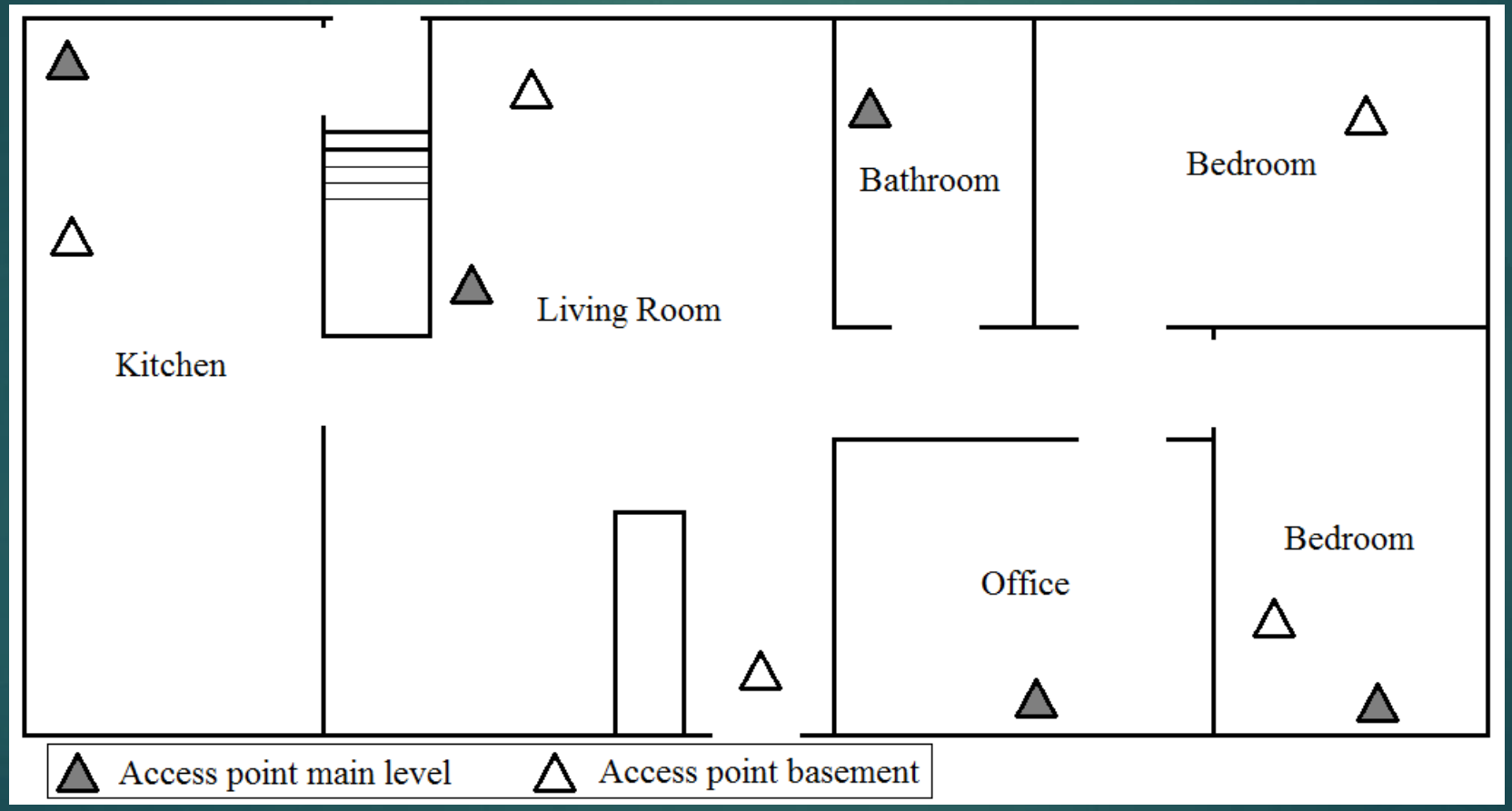


- Central Server
  - Written in Java
  - NetBeans IDE used to write the server
- Arduino Uno Software
  - Written in C
  - Arduino IDE used to write the software and download to the Arduino Uno
- Version Control
  - Git used for main version control
  - BitBucket used to host the Git repository and easily share with entire team

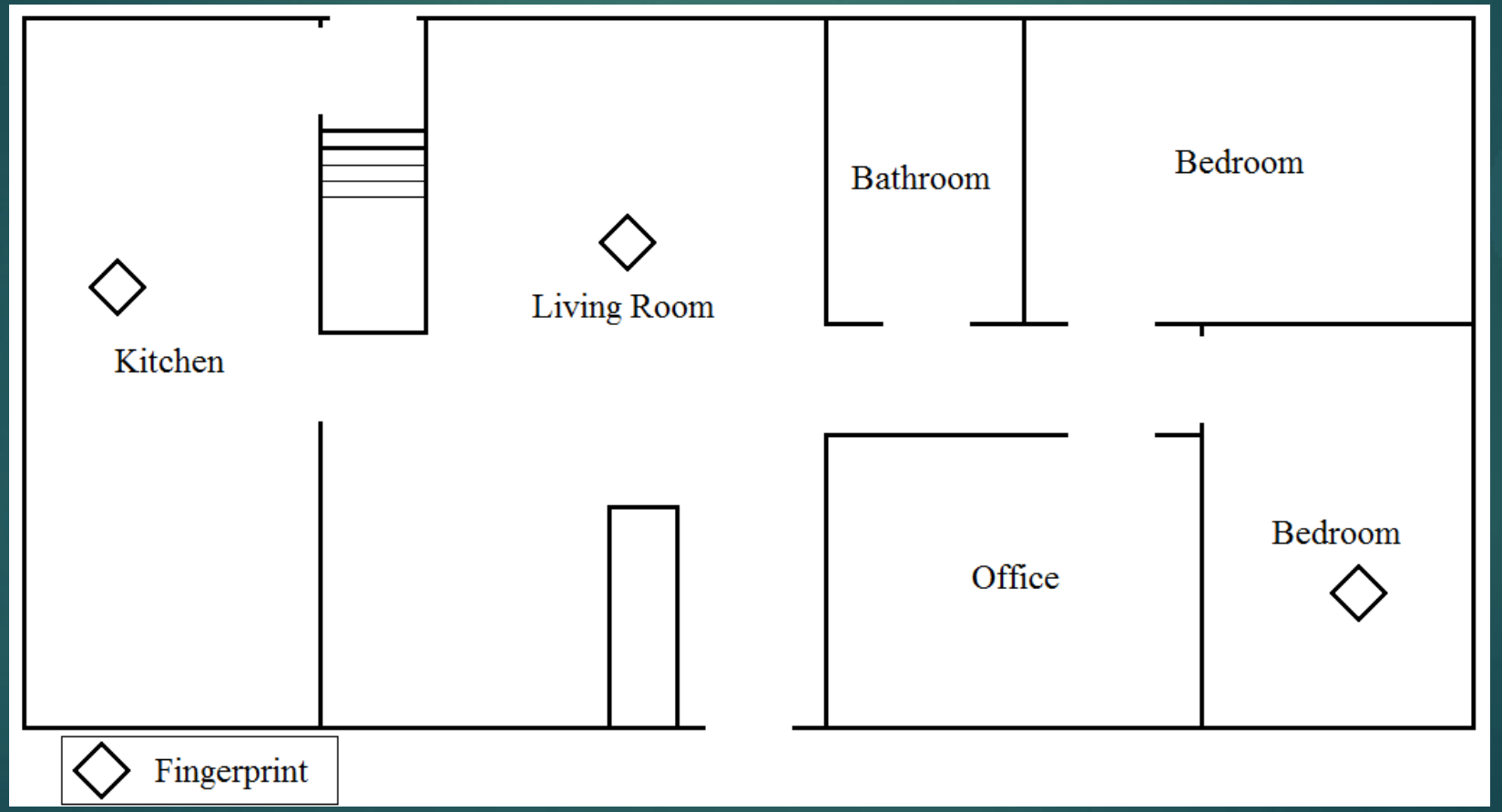
Performance

- Testing the existing Redpin Android app and server
- Parkview
  - No additional access points
  - With additional access points
- House

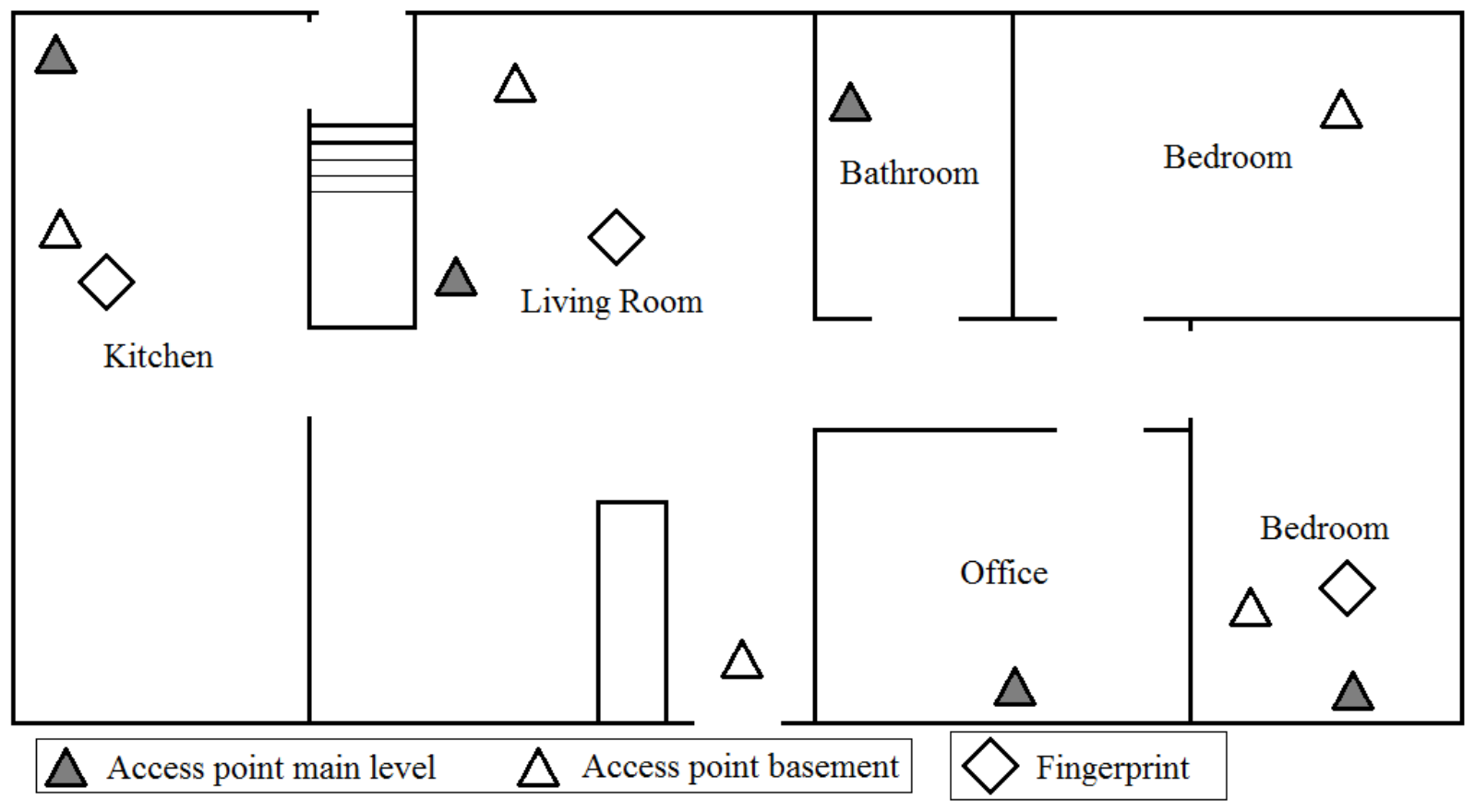
Performance



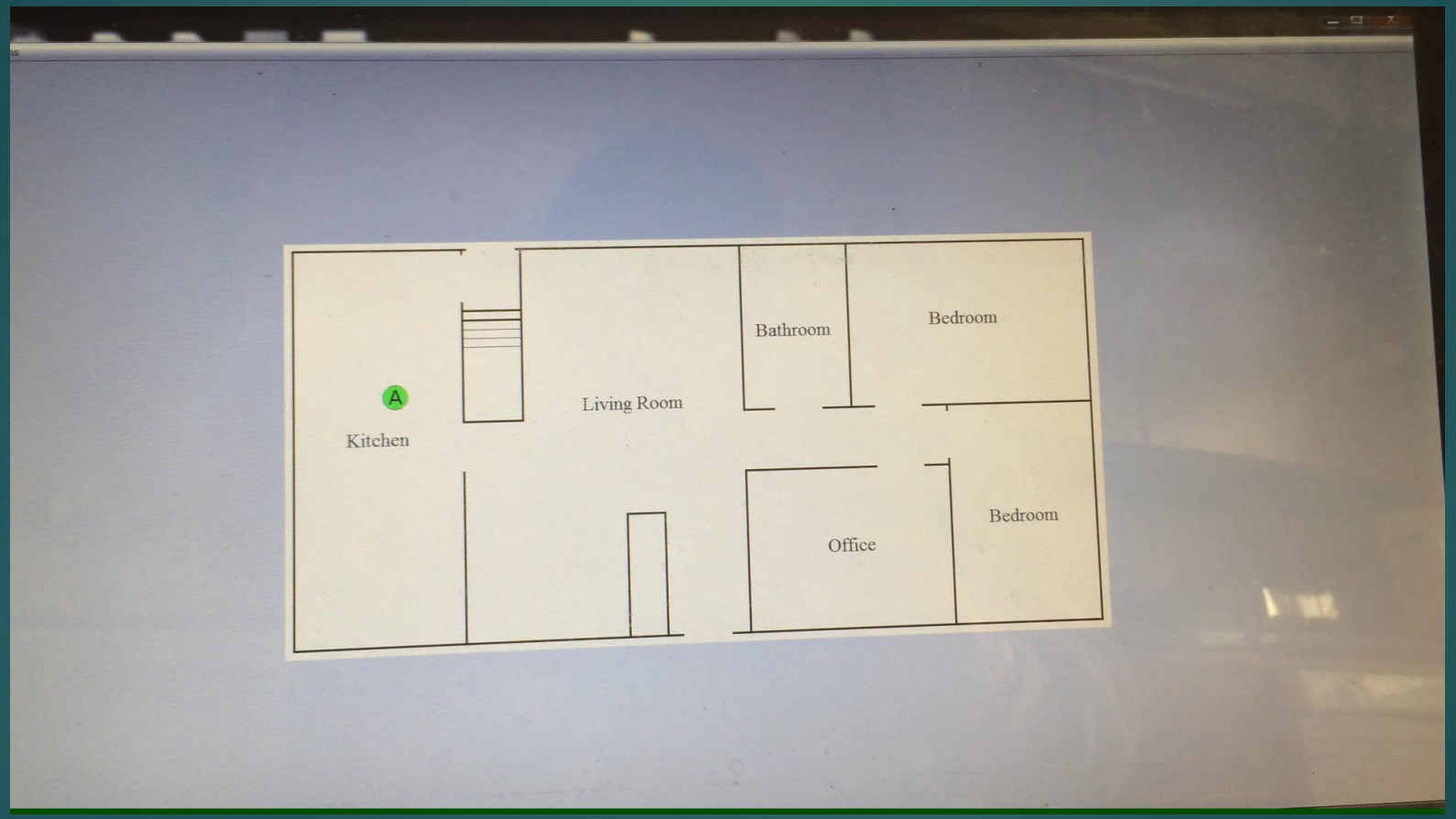
Performance



Performance



Video



Unexpected Problems

- Issues with original microcontroller MSP430 and WiFi shield CC3000
- Arduino WiFi Shield can only locate a maximum of 10 networks
- Redpin triangulation was not consistent at Parkview
- Arduino WiFi Shield cannot return router specific identification (BSSID)

Low Power

Device	Power Consumption (mA)	Duty (8 hrs)	Total (mA)
Atmega328	5.2	100%	5.2
AT32UC3A1256	20	100%	20
HDG104 WLAN (Tx)	178	3.33%	5.93
HDG104 WLAN (Sleep)	0.06	96.67%	0.06
		average current draw	31.19133333
# of transmits per minute	2		
Duration of transmit (sec)	1		
	Low Power		
Device	Power Consumption (mA)	Duty (8 hrs)	Total (mA)
Atmega328 (Idle)	1.2	90.00%	1.2
Atmega328 (Active)	5.2	10.00%	0.52
AT32UC3A1256(Idle)	14	93.33%	14
AT32UC3A1256(Active)	20	6.67%	1.334
HDG104 WLAN (Sleep)	0.06	96.67%	0.06
HDG104 WLAN (Tx)	178	3.33%	5.93
		average current draw	23.04533333
# of transmits per minute	2		
Duration of transmit (sec)	1		

Comparing Performance to Original Specifications

Specification	% Complete
Relatively small device	85%
Low power consumption	50%
Portability	100%
Long operating life	75%
Communicate using 802.11 b/g	100%
Device sends emergency message	50%
Device periodically reports location	100%
Locations of devices shown on map	100%
Inexpensive	50%
Total Percent Complete	75%

Recommendations

- Make the device smaller
  - Design a printed circuit board that only has the necessary components
- Add two-way communication between the devices and Central Server
  - Devices receive messages from central server alerting them of a button press from another device
- Add extra components to the device
  - Liquid-Crystal Display (LCD)
    - Display emergency messages to the user
  - Vibration motor
    - Alert user of an incoming message
- Modify the Arduino WiFi Shield’s firmware and libraries to allow more than 10 access points per scan result and include BSSID in the result

Acknowledgements

- Dr. Ralph Tanner for being our faculty advisor
- Dr. John Kapenga for being our sponsor
- Dr. Damon Miller for teaching ECE 4810 Senior Design I
- Dr. Massood Atashbar for teaching ECE 4820 Senior Design II

Any Questions?