TEXAS STATE®

COMPUTER SCIENCE

JERRY!

SMART DEVICE CONTROLLER

Advanced Software Project

CS5394

Ganta, Aashritha (a_g1374)
Kadkol, Shweta Sudheer (s_k233)
Manika Ganeshbapu, Vishal Kumar (v_m255)
Paliwal, Manvick (m_p475)
Chada, Meghana (m_c754)

1. Overview:

This report records the detailed description of the voice-based device-controller, its functionalities, and features. It also points out the system constraints, hardware and software interface, and a communication interface.

A smart device is an electronic device, generally connected to other devices or networks via different wireless protocols such as Bluetooth, NFC, Wi-Fi, LiFi, 3G, etc., that can operate to some extent interactively and autonomously. Lately, smart devices have been synonymous with anything that's network enabled. Whether, that is lights (Phillips Hue), thermostats (Nest), or even toothbrushes (Prophix).

The idea behind 'Jerry!' is to build a voice-command based controller which manages devices like lights and fans etc. making it interactive and efficient at the same time.

A web interface which picks voice-commands → performs the given instruction → states what is being done is designed to make the controller easy to use.

2. Introduction:

2.1 Product:

The voice-based device-controller system is a combination of hardware and software that will provide a user the means to automate device-management. The system will allow a user to create and access profiles that correspond to specific routines. For each voice-command, the system must accurately interact and perform the desired operation(s). The system also has a dashboard which lets the user view his/her list of devices and operate them with GUI as well.

2.2 Circuit:

2.2. A COMPONENTS

A. ESP 8266
B. 5v relay switch
C. Router
D. Connecting wires
E. PCB
F. 5v power supply (adapter will do)
G. AMS 1117 dc voltage regulator
H. Soldering iron
I. Header pins
J. FTDI to USB adapter

2.2. B Definitions:

A. UI/GUI- User Interface and Graphical User Interface respectively.

B. ESP8266 – is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability produced by manufacturer Espressif Systems.

C. 5v relay switch- An electrically operated switch used mostly when in need of controlling a circuit by a separate low-power signal.

D. Router – is a networking device that forwards data packets between computer networks.

E. PCB -Printed Circuit Board mechanically supports and electronically connects electronic components. It usually fastens the components.

F. AMS 1117 dc voltage regulator – Advanced Monolithic Systems is a component used to automatically maintain a constant voltage level

G. FTDI - Future Technology Devices International is a semi-conductor device company

H. USB adapter – Universal Serial Bus adapter eliminates the need to hardwire the computer to a router or other network device.

I. FSM – Finite State Machine - A finite state machine (sometimes called a finite state automaton) is a computation model that can be implemented with hardware or software and can be used to simulate sequential logic and some computer programs.

J. GPIO – General Purpose Input/output – a standard set of pins available on microprocessors which allow communication between various peripheral

K. SSID- Service set Identifier-An SSID (service set identifier) is the primary name associated with an 802.11 wireless local area network (WLAN) including home networks and public hotspots. Client devices use this name to identify and join wireless networks.

3. Requirements:

Jerry! as a project was approached in an agile environment and went through a Software Development Life Cycle. A requirement gathering was an important phase in that direction. Software Requirement Specifications for the project is available in dropbox as well as Git.

4. System Operations:

The system should have default setting as follows:

1. ESP8266 GPIO – LOW
   A. ESP8266

Jerry! shall be implemented with an ESP8266 Version 01

a. 1MB Flash
b. Built-in TCP / IP protocol stack
c. 32-bit micro controller
d. 80 MHz
f. WIFI enabled

2. ESP 8266 can be programmed using:

a. Arduino Software

3. Components shall be installed on PCB

4. Connections

a. 5v Power supply connects to Relay Switch and AMS1117.
b. AMS1117 gives 3.3v output to ESP8266 Module.
c. ESP GPIO Pin 2 connects to Relay Switch Control Pin.

5. Relay Switch – OFF

6. Connection:

a. Connected to Home router.
b. Connected to cloud server.
c. Actively listening for any actions.

4.2 INTERFACE OPERATIONS

Jerry's web interface defines the requests and the skills can handle (device directives) and the words users say to invoke those requests (utterances).

A. All voice interactions with the user are handled by the Web application Interface.
B. You supply one or more reliable content feeds in RSS or JSON format. The content can be audio content that the device plays to the customer or text content that device reads to the customer.
C. The code then handles the request appropriately. I.e. it will handle the request in sequence that the user provides.
D. With respect to GUI, the dashboard is designed to make it as user friendly as possible and to keep it simple overall. Micro framework like flask makes the application light and accommodative of most browsers.

5. Use Cases:

5.1 Joining Wireless Network

Name of Use Case:	Connecting to a valid SSID
Brief Description:	The user searches for WIFI connection and logs in to access Jerry!
Pre-Conditions:	Computational device with a standard browser
Post-Conditions:	The system can be accessed via a browser
Flow of Events:	1. User selects a valid SSID and correct credentials 2. Computational device is connected to Internet 3. The Homepage of Jerry! Is displayed to be either logged in or sign up
Alternate Flow and Exceptions:	1. A user interaction message has been displayed. 2. The user performs the appropriate action, i.e. connect to the network 3. The screen returns to the main menu.
Alternate Flow and Exceptions:	1. Credentials aren't valid or the WIFI connection isn't strong. 2. The user choses a response to the error- rectify credentials or join another strong network. 3. The system responds and goes back to the Homepage of Jerry!

5.2 Creating User Profile

Name of Use Case:	Creating user profile
Brief Description:	The user creates his/her profile to achieve customization
Pre-Conditions:	Connected to the network and on the homepage of Jerry!
Post-Conditions:	The information entered by the user is saved and stored successfully to be accessed any time in future
Flow of Events:	4. User enters relevant information to each attribute 5. Screen displays the information. 6. The user reviews the information and saves it to access in future.

Alternate Flow and Exceptions:	4. A user interaction message has been displayed. 5. The user performs the appropriate action, i.e. rectify the information or provide necessary information from must-have fields. 6. The screen returns to the main menu.

5.3 Manage Profile

Name of Use Case:	Manage Profile
Brief Description:	The user updates credentials; edits profile information; Logs in and out
Pre-Conditions:	Should have a user profile
Post-Conditions:	The information entered by the user is saved and stored successfully to be accessed any time in future. Safely logging in and out of the system
Flow of Events:	7. User enters updated information or new username-password 8. Screen displays the information to be reviewed. 9. The user reviews the information and safely logs out of the system
Alternate Flow and Exceptions:	7. A user interaction message has been displayed. 8. The user performs the appropriate action, i.e. rectify the information or provide necessary information from must-have fields. 9. The screen returns to the main menu.

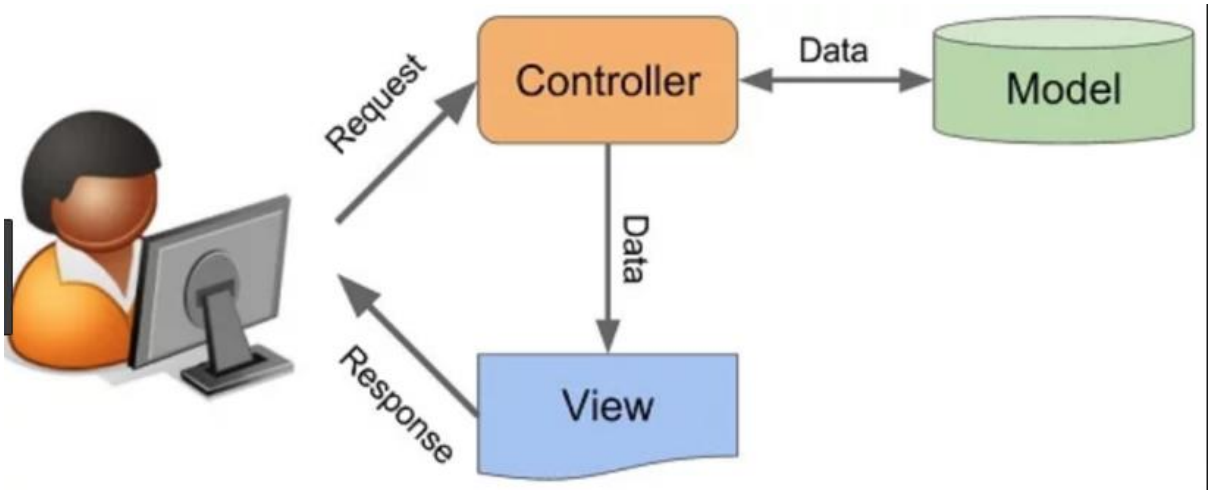
5.4 GUI based device management

Name of Use Case:	Interacting with the system using GUI
Brief Description:	Using dashboard to manage devices
Pre-Conditions:	Should have a user profile and internet connection
Post-Conditions:	Jerry performs the requested task and changes on the dashboard accordingly.
Flow of Events:	10. Use the Graphical interface to target a device 11. Instruct to turn the device on/off 12. Jerry! does accordingly
Alternate Flow and Exceptions:	10. User is informed of the error 11. Wait to be restored 12. Switch to voice-based controller

5.5 Voice based device management

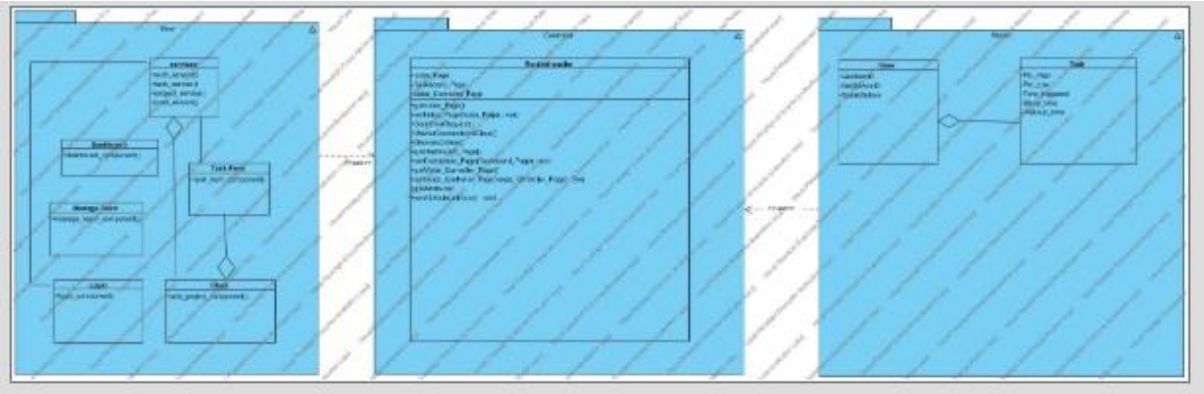
Name of Use Case:	Interacting with the system using specific keywords and voice
Brief Description:	Voice-based controlling
Pre-Conditions:	Should have a user profile and internet connection
Post-Conditions:	Jerry performs the requested task and informs accordingly.
Flow of Events:	13. Use the customized voice-commands to target a device 14. Instruct to turn the device on/off 15. Jerry! does accordingly
Alternate Flow and Exceptions:	13. User is informed of the error 14. Wait to be restored 15. Switch to GUI-based controller

6. Framework



The project follows a Model-View-Controller Framework as outlined above. Requests from User, be it voice commands or dashboard related, are sent to controller which in turn interacts with Model (Circuit). The responses are sent to User Interface (View) which is available for user to see.

6.1 Class Diagram



7. GIT:

Since Jerry! was designed and developed in an agile environment , it made sense to create a version control system of GIT and track our progress as a team.
https://git.txstate.edu/v-m255 is the URL to access the system and it has 5 branches as of now. We plan to add more branches and commits as we achieve the Future work.

8. Testing

Since our Model-View-Controller is based on both Hardware and Software Components, the testing phase was initiated on Hardware from the very beginning of the project.
Initially we worked with relay switch which restricted us to just one device. However we wanted to simulate two rooms, garage and the yard hence we had to use Light Emitting Diodes. Monetary and Time constraints means we couldn't use sensors and make it more automated.

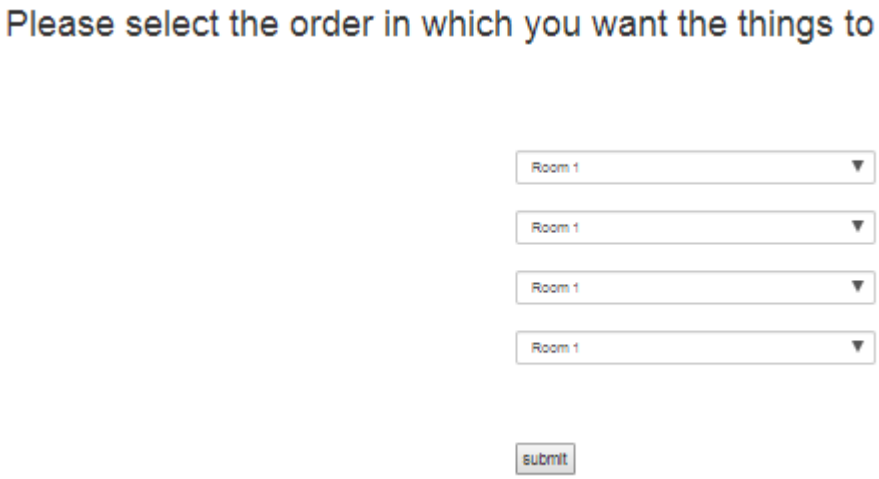
With respect to software testing , We decided certain business rules based on our design constraints and made sure to include failure routines and exceptions in case of edge cases.

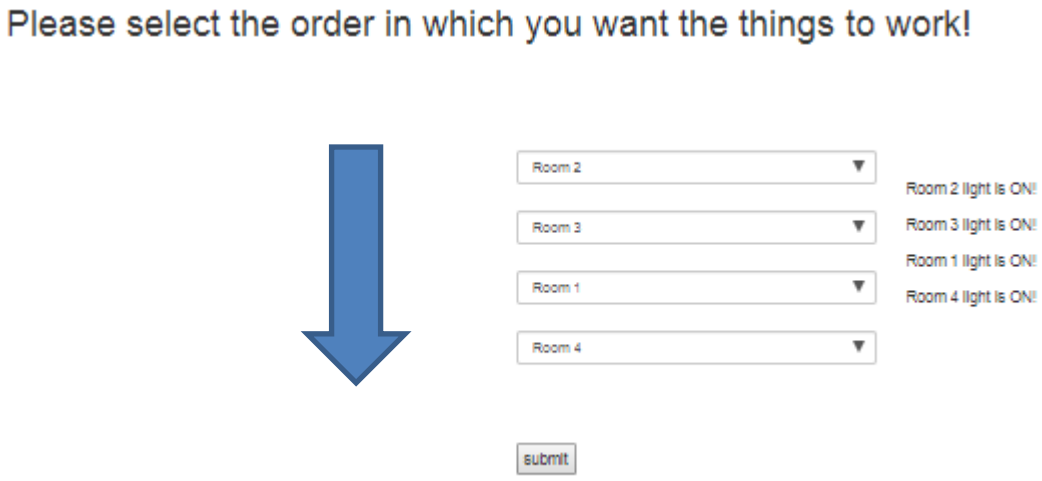
8.1 Failure Routines:

a. In scenarios like power failure or internet connection failure, the dashboard lists all the devices as offline and automatically shows them online when the desired settings are restored.
b. Currently, the system cannot operate in cases of power or internet connection failure. During this period, the voice-command settings will be turned-off.

9. FSM-like Routine Simulation

Courtesy, Dr. Podorohzny's suggestions we also implemented a Finite State Machine like routine. This routine is a simulation at this point because of Cost and Hardware constraints. The 4 routines are step-by-step visual simulation of device management based on their priorities. The user selects the priority with which he/she wants the services triggered. This makes Jerry! in a way ubiquitous and controls more ground.





10. Other Features

10.1 Performance Requirements Met:

1. Multiple devices can be managed at the same time.

2. Response time is less than a minute under normal conditions

3. Accents are considered.

4. Interactive system.

10.2 Non Performance Requirements Met:

1. Security – Creating a user profile lets the user safely log in and out of the system.

2. Accessibility – The very purpose of the system is to make device-management more easy and accessible.

3. Memory constraints – Since the system isn't offline, memory management is not an issue.

4. Time to respond – Is not more than a minute when the internet and power connections are fine.

5. Ease of Use – GUI and voice-commands are kept simple to make it easy to use.

6. Adaptability – The system can adapt to various kinds of devices and can be extended to an interface which is mobile friendly.

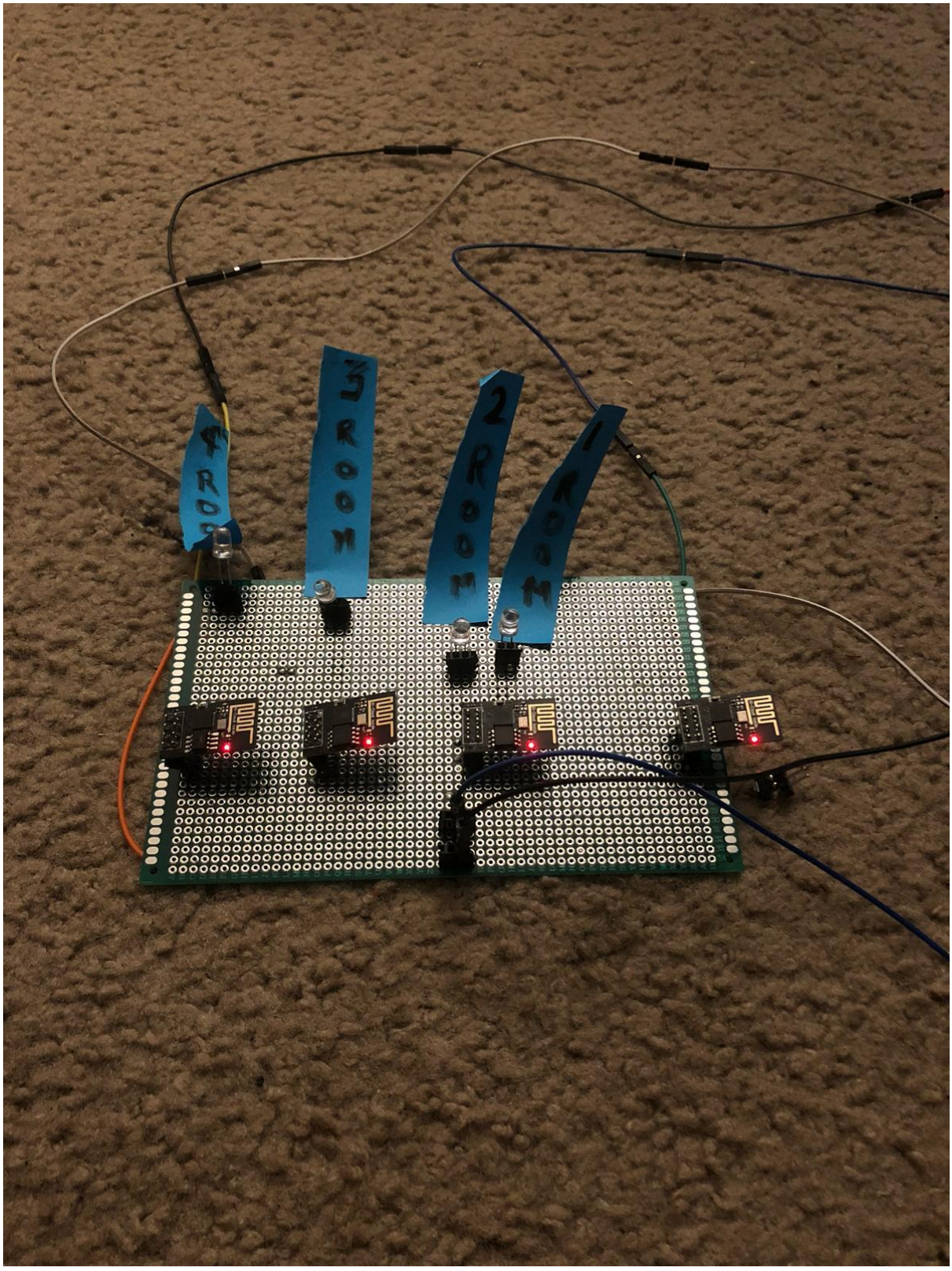
10.3 Weather, Maps and News:

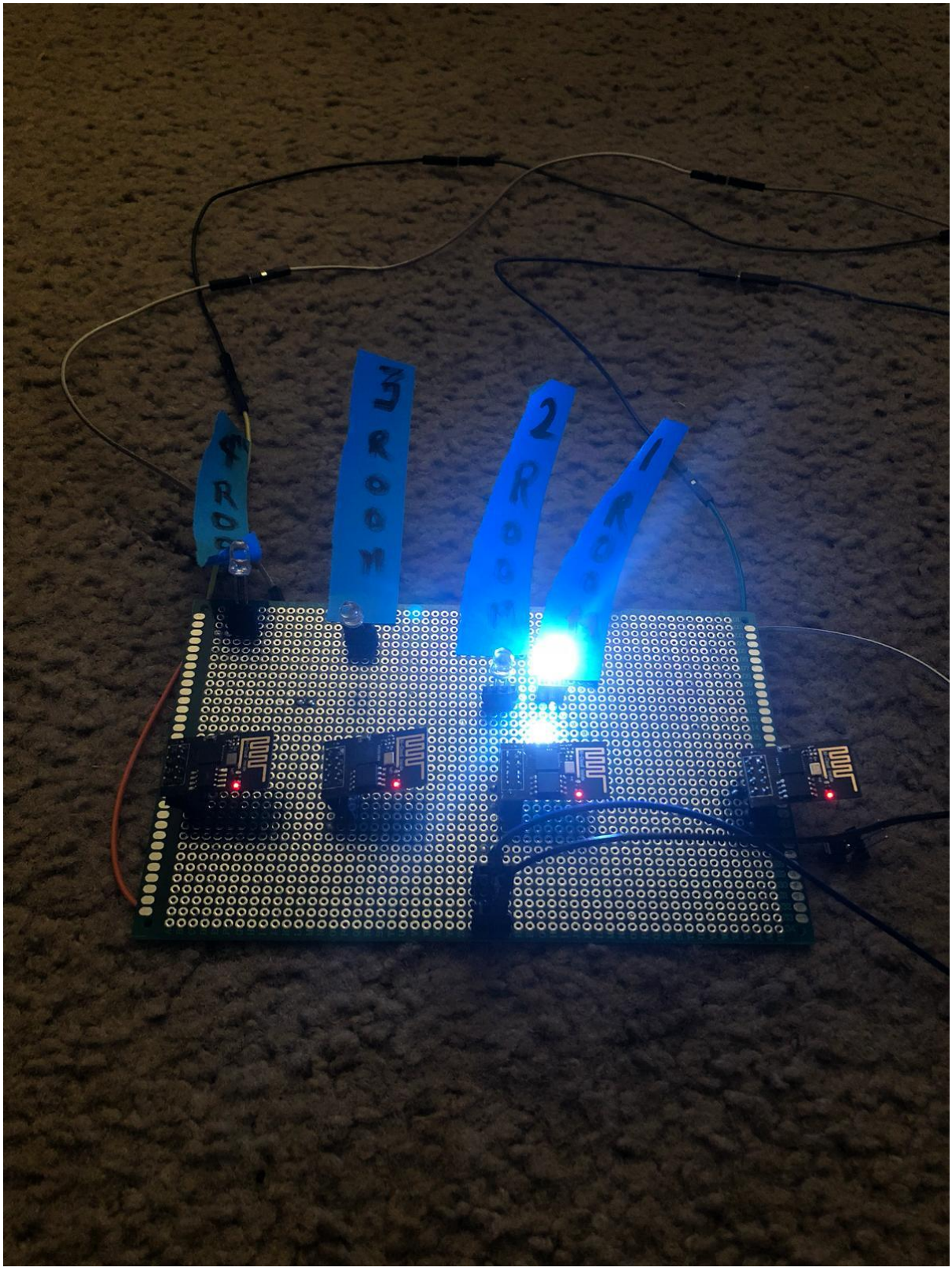
The Homepage of Jerry provides sources to weather, maps and news. This is an effort to make the user's morning routines accessible at one stop point. Due to time constraints, at this point, these features redirect you to the concerned website.

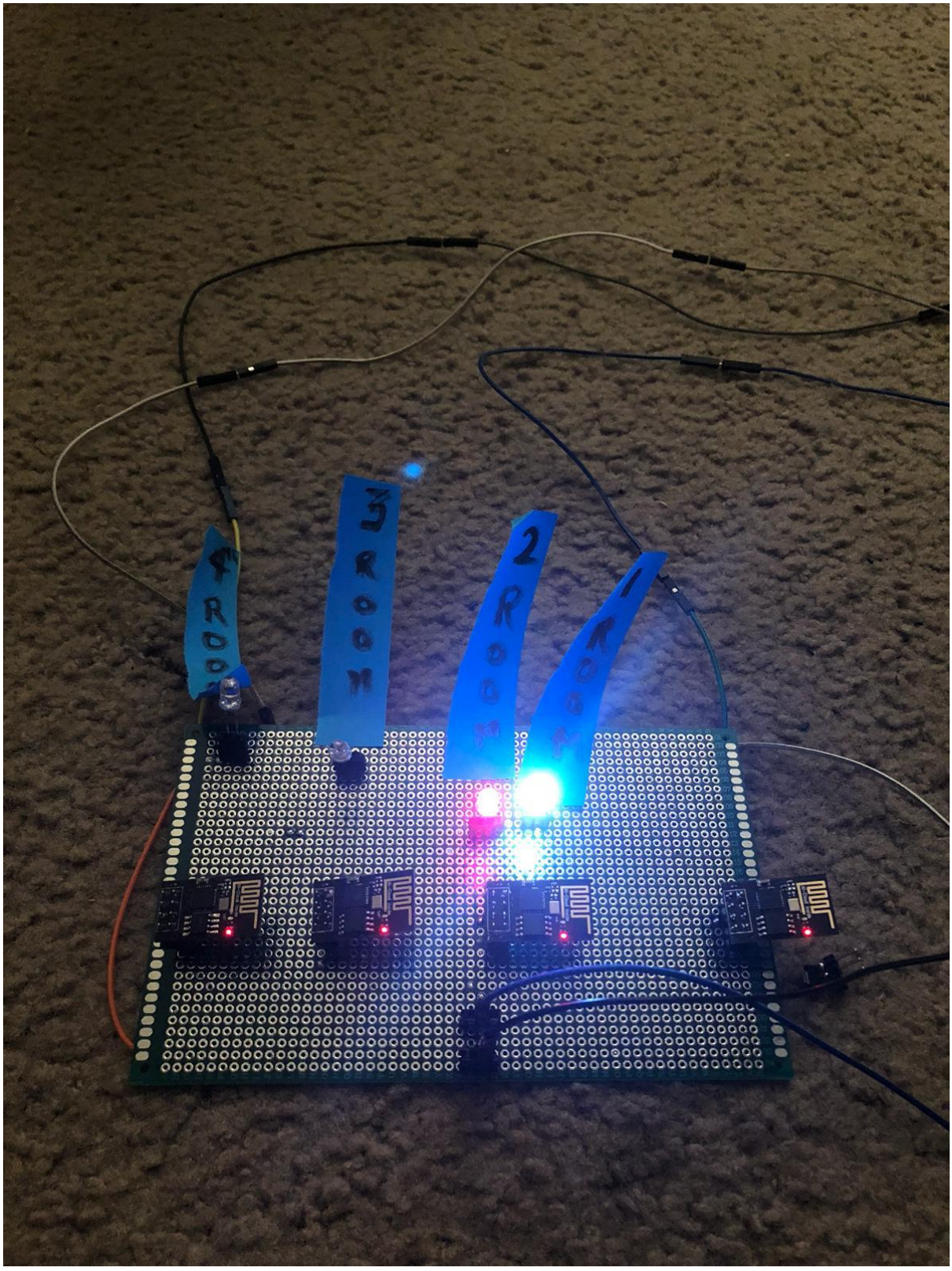
11. Verification and Validation

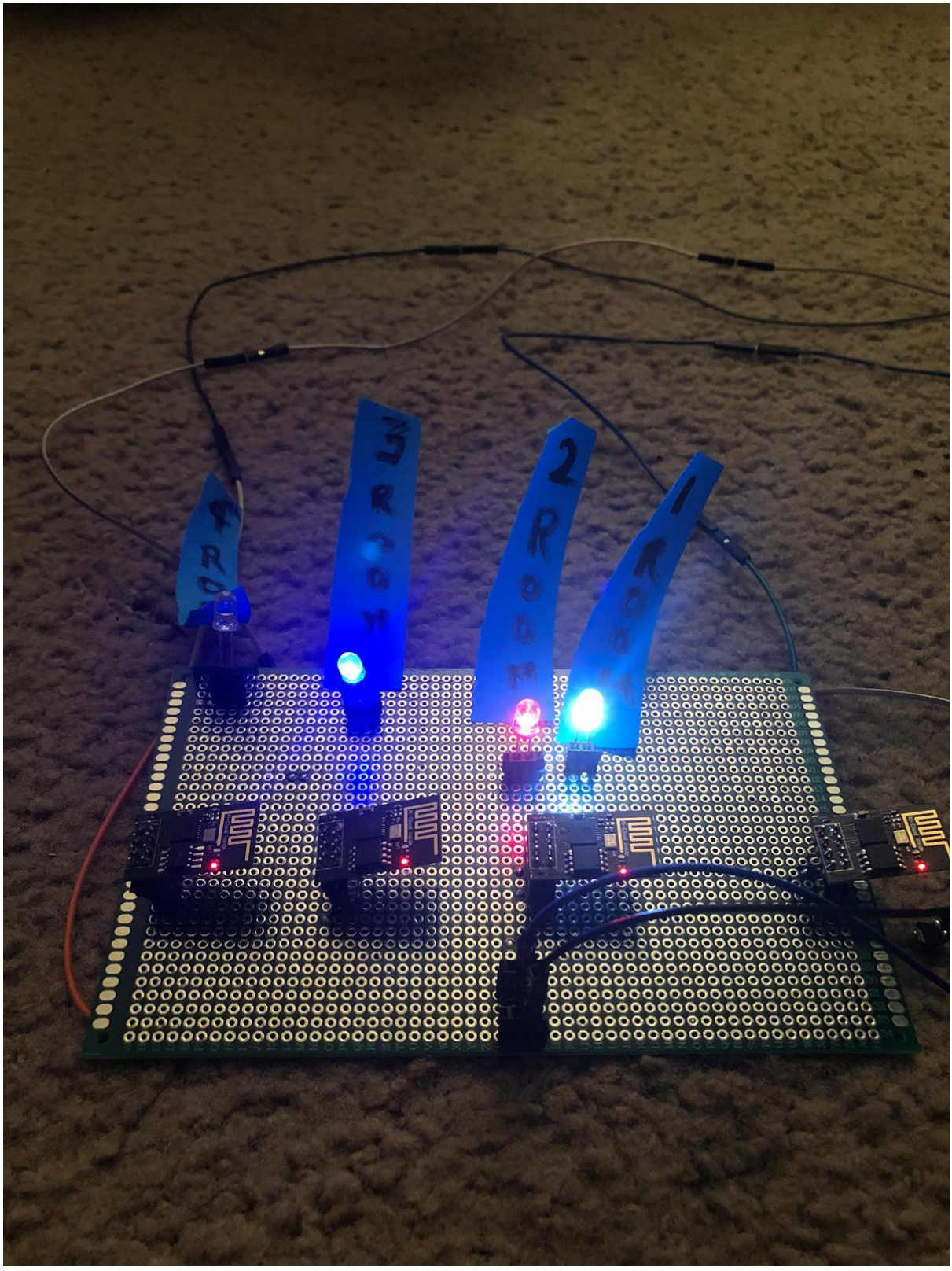
The project description, design, class diagram and code are all updated as commits on the Git provided. These logs were used in the Verification phase of the Project.
https://git.txstate.edu/v-m255/ProjectJERRY/blob/master/projectJerry_Functional.mp4 is a demonstration video which facilitates the validation of the product at both hardware and software level. Live demonstration was arranged for the class but this video serves to help the users who view the product at later times.

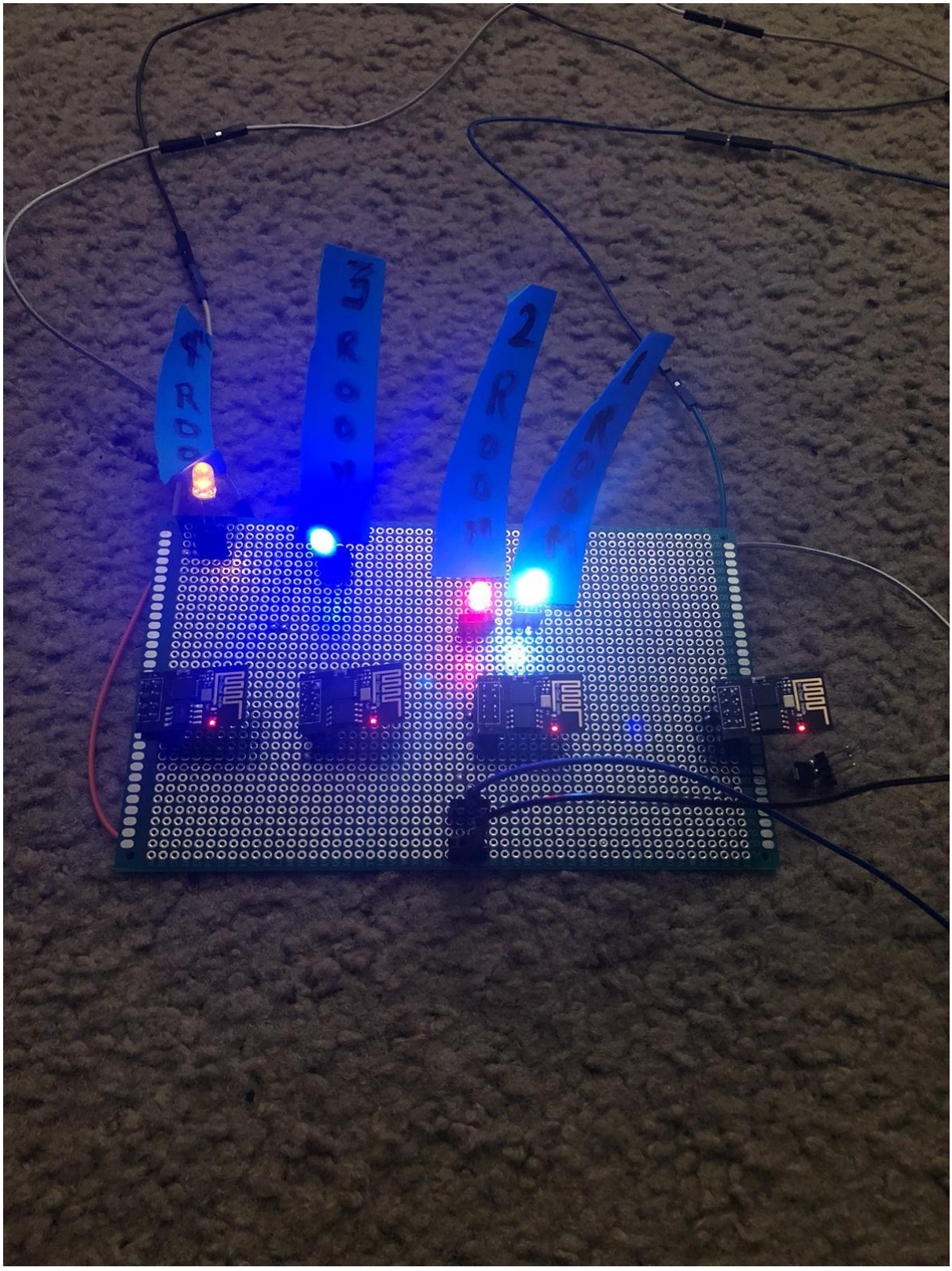
12. Results


3 ROOM
2 ROOM
1 ROOM


3
ROOM
2
ROOM
1
ROOM


3 ROOM
2 ROOM
1 ROOM


3
2
1



13. Future Work

a. Actual implementation of FSM-like routine instead of simulation to achieve smart home.

b. Fetch results from trusted sources with respect to news, weather and maps to present a Gist to the users
c. Make the product, especially the dashboard more mobile-friendly.
d. Limit the constraints on exceptions.

14. Bibliography

[1] Smart devices and soft controllers - C.W. Thompson Arkansas Univ., USA
[2] Smart objects as building blocks for the Internet of things - Gerd Kortuem; Fahim Kawsar; Vasughi Sundramoorthy; Daniel Fitton
[3] Building smart services for smart home - S. Chemishkian
[4] Performance Analysis and Comparison on Energy Storage Devices for Smart Building Energy Management-Zhanbo Xu, MOE KLINNS Lab ofXi'an Jiaotong University, Xi'an, China; Xiaohong Guan; Qing-Shan Jia; Jiang Wu; Dai Wang; Siyun Chen.
[5] Building intelligent environments with Smart-Its L.E. Holmquist; H.W. Gellersen; G. Kortuem; S. Antifakos; F. Michahelles; B. Schiele; M. Beigl; R. Maze
>> http://gitimmersion.com/
>> https://www.instructables.com/id/Smart-Voice-Controller/