Team 5 Project Plan Documentation

Prepared for: Prof. Healy

Oregon Institute of Technology

Prepared by Hayden Hutsell, Zak Rowland, James Rountree, and Beto Estrada

December 9, 2019

Table of Contents

Modules Within the Project

Within the electronic directory board's code and hardware there are four distinct modules: the Wi-Fi module, the shift register RGB LED module, the shift register infrared sensor and LED module, and the localhost website.

Wi-Fi Module

The Wi-Fi module communicates with the localhost website to receive the current schedule information.

- Will receive the status for all rooms from the localhost website.
- Will be able to retrieve the current time.

Shift Register RGB LED Module

Based on the input received from the Wi-Fi module, it will use the shift out library in the Arduino library to interface with the serial-in to parallel-out shift registers. The output will light the RGB LED to display the current status of the room (booked, open, canceled).

- Will need to process the input from the Wi-Fi module and send it to the shift registers.
- The shift registers will be powered from an external 5-volt power supply.
    - The external power supply is sourcing current to the RGB's, not the ESP-32.
- Utilizes Arduino's shiftOut library.

Modules Within the Project Continued

Shift Register Infrared Sensor and LED Module

Using a parallel-in to serial-out shift register, the shift register will send data to the ESP-32 showing which motion sensor is detecting motion. When motion is detected, a timer for five minutes is set, and its corresponding LED is turned on (refer to figure 3, figure 5). When motion is detected while the timer is active, the timer is reset to five minutes. When the timer runs out, the LED turns off. Implementation of the LED toggling is done through a serial-in to parallel-out shift register, like the previous module.

- Like the previous module, the shift registers will be externally powered.
- Parallel-in to serial-out for the motion sensors, and serial-in to parallel-out for the 8 LEDs.
- Utilizes Arduino's shiftOut and shiftIn libraries.

Local Host Website

The localhost website will hold the complete schedule for a week for each of the lab rooms. The data from the schedule will be retrieved by the ESP-32 via Wi-Fi. Each hour will be able to be edited in case of alteration or cancellations. It will also have some security measures in place to make sure only authorized individuals can edit the schedule.

- Will be able to be modified by those with a login.
- Each room has a base schedule, and each hour block can be edited.

Detailed Diagram Descriptions

Wi-Fi Diagram

The Wi-Fi diagram and functionality are simple. It retrieves data from the localhost website, and it retrieves the time from an NTP server (figure 7).

Shift Register RGB LED Diagram

This diagram deals with the 595N shift registers and LEDs. We are using a version of 595N shift register that can supply more current to the output pins, so the LED's receiving a steady flow of power. For the RGB shift registers, there are three shift registers daisy chained together (upper half of figure 4). The Arduino is connected to the 595N on three pins: serial in (the data that will be shifted to the outputs), SRCK (shift register clock), and RCK (register clock, basically acts as a latch that will send the outputs their respective value after the serial data has been received). When the 8th bit is shifted out, it goes to the first bit of the next shift register.

Shift Register Infrared Sensor and LED Diagram

This is the lower half of figure 4. This implementation is similar to the shift register RGB LED diagram (upper half of figure 4). It is the same shift register, except there is only one, and there are 8 single color LEDs. Basically, a simpler version of the shift register RGB LED diagram.

Local Host Website Diagram

The localhost website diagram (figure 8) has a simple flow of information. The user will log in to the website before they can edit the schedule information. Once the user is logged in, they may edit any lab room's schedule. After they edit any schedule, when the ESP-32 polls the website to check if anything has changed (done every minute), the changes will be reflected on the RGB LEDs.

Test Plans

Shift Register RGB LED Module

This module contains the 595N shift registers and the RBG LEDs. The 595N shift registers have already been tested along with making sure there is no problem with LEDs flickering when their status is being changed. That was solved by adding a capacitor on the voltage-in to the shift register. However, we will need to test the different model of 595N we are getting, as it can sink more current than the 595N we tested. Initially, we will test the shift registers with the RGBs to make sure the wiring is correct, the RGBs work, and the functionality is the same as the previous 595N we tested. The final module test will include simulating a change in the schedule block and sending different bits to the registers to see if the LEDs reflect accurate information.

Shift Register Infrared Sensor and LED Module

This module will contain the CD4012B shift register, the passive IR sensors (PIR), and the single-colored LEDs to signify movement. The PIR sensors will need to be tested for how sensitive they are to movement and whether the equipment in the rooms cause the sensors to give off a false-positive result. The testing of the module itself will include lighting an LED when the motion sensor is triggered.

Wi-Fi Module

This module contains the ESP-32 Arduino as it has the WI-FI chip integrated into the board. The testing of the module is to make sure it can retrieve information wirelessly and accurately. We have already tested our ESP-32's by connecting to a Wi-Fi network and retrieving the current time through an NTP server. Final testing includes retrieving data from the localhost website and outputting it to serial out so we can verify the data is accurate.

Local Host Website

This module is software-based, so testing will be simpler. The ESP-32 must be able to retrieve the data from the website accurately. This can be tested by outputting the data retrieved from the website to putty, using the serial out function on the Arduino. This way we can verify the data retrieved is accurate before we move on to lighting up the LEDs with the retrieved data. The main testing of the module will include, login security, updating and setting up the part that will hold the schedule, and making sure that the module sends out the correct information based on the schedule.

Test Plans Continued

Module Integration

After the individual parts and modules have been tested and finished the module will have to be integrated into each other for the entire project to work. The first integration will be integrating the LEDs and shift registers. The second integration will be of the two shift registers together testing if the Arduino can handle all the pins needed for the project to work. The third integration will be the final integration. The shift registers will be integrated with the Wi-Fi module, so the RGBs will be controlled by the data from the localhost website.

Updated Parts List

The major parts needed for the project are the ESP-32 microcontroller, the PIR sensors, and the router to communicate with the localhost website. Additionally, we will need RGB LEDs, LEDs, resistors, shift registers, a power supply, breadboard-friendly barrel connectors, and housing. Those are the router which as noted below was donated by Hayden and the ESP-32 microcontroller, which was ordered early to test the wireless capabilities. The remaining parts will be ordered next week, besides the housing, which will be ordered after our prototype is constructed.

The biggest change in our parts list was our choice with the Arduino. We initially decided with the Arduino UNO Wi-Fi, but after testing the ESP-32 and optimizing how our LEDs will be powered, we determined the ESP-32 would work for our project. Not only do we get two ESP-32's for the cost, but we also save $30. There have been minor changes with powering the LEDs, from MOSFET circuits to now an external power supply to power the shift registers, which source the current to power the LEDs. Cheaper, and easier to construct.

Software Specifications

Shift Register RGB LED Module

This module will need to take the information sent from the localhost website and translate it into the correct information needed to run the RGB LEDs. Since this will be using the shift registers on RGBs, the data will need to be translated into what color the RGB needs to be and shift that out into the shift registers. This module will also poll for changes every hour for the schedule changes, and it will also poll every minute for changes on the current schedule if a professor cancels a class as shown in figure 2.

Shift Register Infrared Sensor and LED Module

This module will take the data from the PIR sensors and convert it into a usable signal to control whether a single-color LED is on or off. The sensitivity of the PIR sensor will need to be tested and adjusted to prevent false positives. The module will poll the status of the PIR sensor every 5 minutes to test whether there is activity in a room. This value is slow enough (five minutes) that not moving for a while will not cause the system to show that there is no one in the room as shown in figure 3. Wiring is specified in figure 5.

Wi-Fi Module

This module will retrieve the information from the website and transmit it to the microcontroller for it to decode. The information transmitted will be a stream of a single bit at a time that will control the RGB LEDs. This data will be states of the LEDs such as one (001) equaling red when decoded by the microcontroller and so on for the other color states being used by the RGB LEDs. This module will also be responsible for sending the time to the microcontroller so that it can determine when a minute or an hour has passed.

Local Host Website

The module will hold the information about the schedule for an entire week. On an hour to hour basis, the microcontroller will retrieve the schedule information through the Wi-Fi module. Each hour will be able to be edited in case of alteration or cancellations. If there is a change within the current hour block, the new data will be retrieved when the Wi-Fi module retrieves data again every minute (figure 2) to be reflected on the board. It will also have some security measures in place, such as a login, to make sure only authorized individuals can edit the schedule.

Individual Team Member Assignments

- **Hayden**
  - Ordering and receiving parts.
  - Tested 595N shift registers.
  - RGB state code.
  - LED and Shift register testing and implementation.
  - Interfacing the ESP-32 to the website.
  - Testing the ESP-32 and website interface.
  - Final implementation of all modules.
- **Zak**
  - Ordering parts.
  - Tested ESP-32 Wi-Fi functionality.
  - Motion sensor coding and testing.
  - LED and Shift register testing and implementation.
  - Interfacing the ESP-32 to the website.
  - Testing the ESP-32 and website interface.
  - Final implementation of all modules.
- **James**
  - Developing localhost website to interface to the board.
  - Interfacing the ESP-32 to the localhost website.
  - Testing the ESP-32 and website interface.
  - Final project implementation.
- **Beto**
  - RGB state code.
  - Motion sensor coding and testing.
  - Final project implementation.

Figure 1. Our updated table of parts to purchase.

Part	Qty.	Cost/unit	Total cost
ESP32 Arduino (2x)	1	$13.27	$13.27
Wireless router	1	Donated by Hayden	$0.00
RGB LEDs (10pk.)	1	$5.95	$5.95
Blue LEDs (25pk.)	1	$5.95	$5.95
Assorted resistors	1	$5.99	$5.99
595 Shift registers	7	$1.14	$7.98
CD4021B shift register	3	$0.43	$1.29
Passive IR sensors (5pk.)	2	$13.99	$27.98
5V/2A+ Power supply	1	$13.90	$13.90
Bread board friendly 2.1mm barrel connectors (10x)	1	$8.99	$8.99
Housing estimate	-	$55.00	$55.00
Total	-	-	$146.30

Figure 2. State Machine for RGB and main State Machine

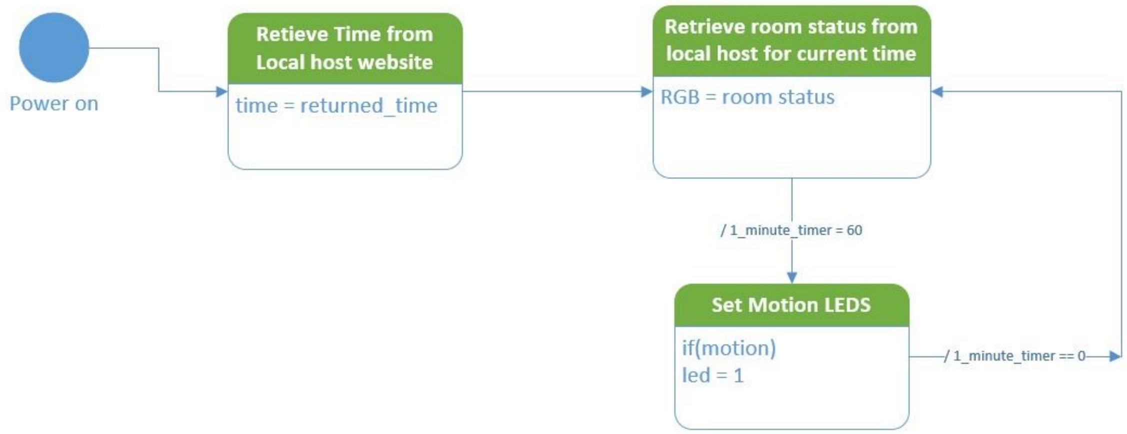

Figure 3. State Machine for PIR Sensors

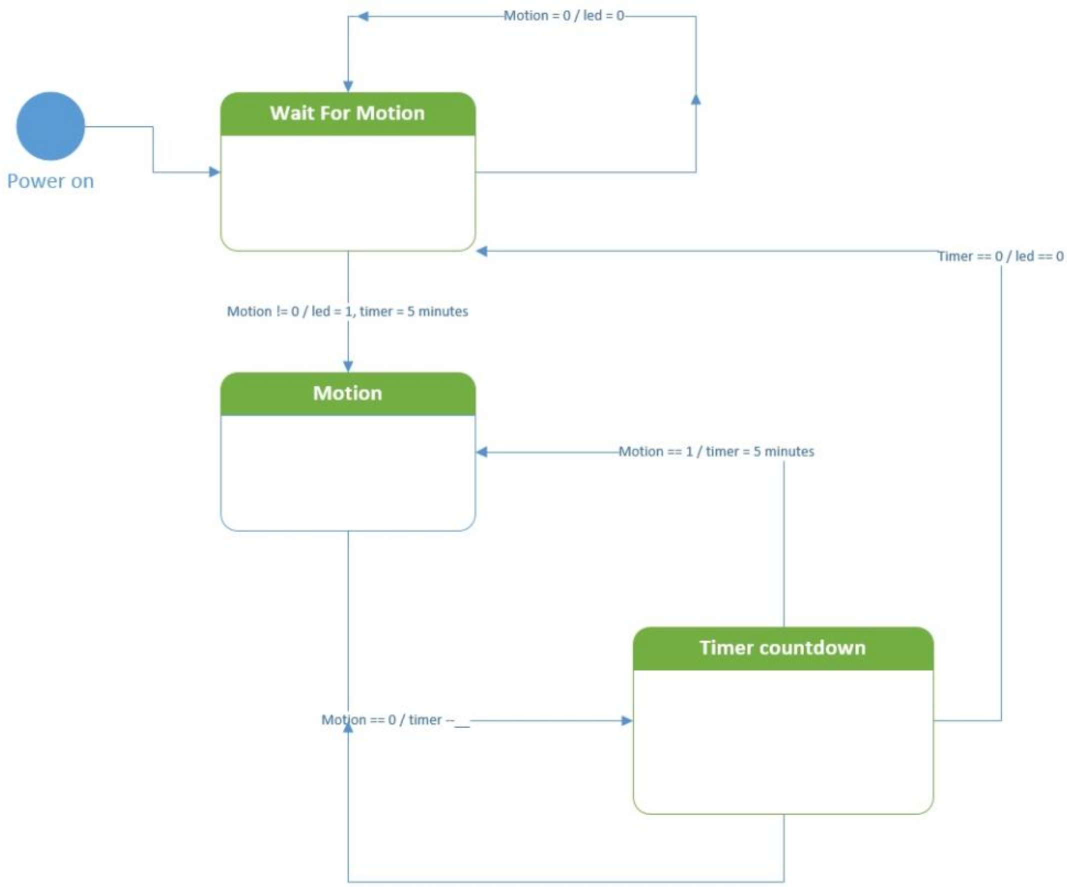

Figure 4. Schematic of the RGB LEDs, LEDs, and Shift Registers

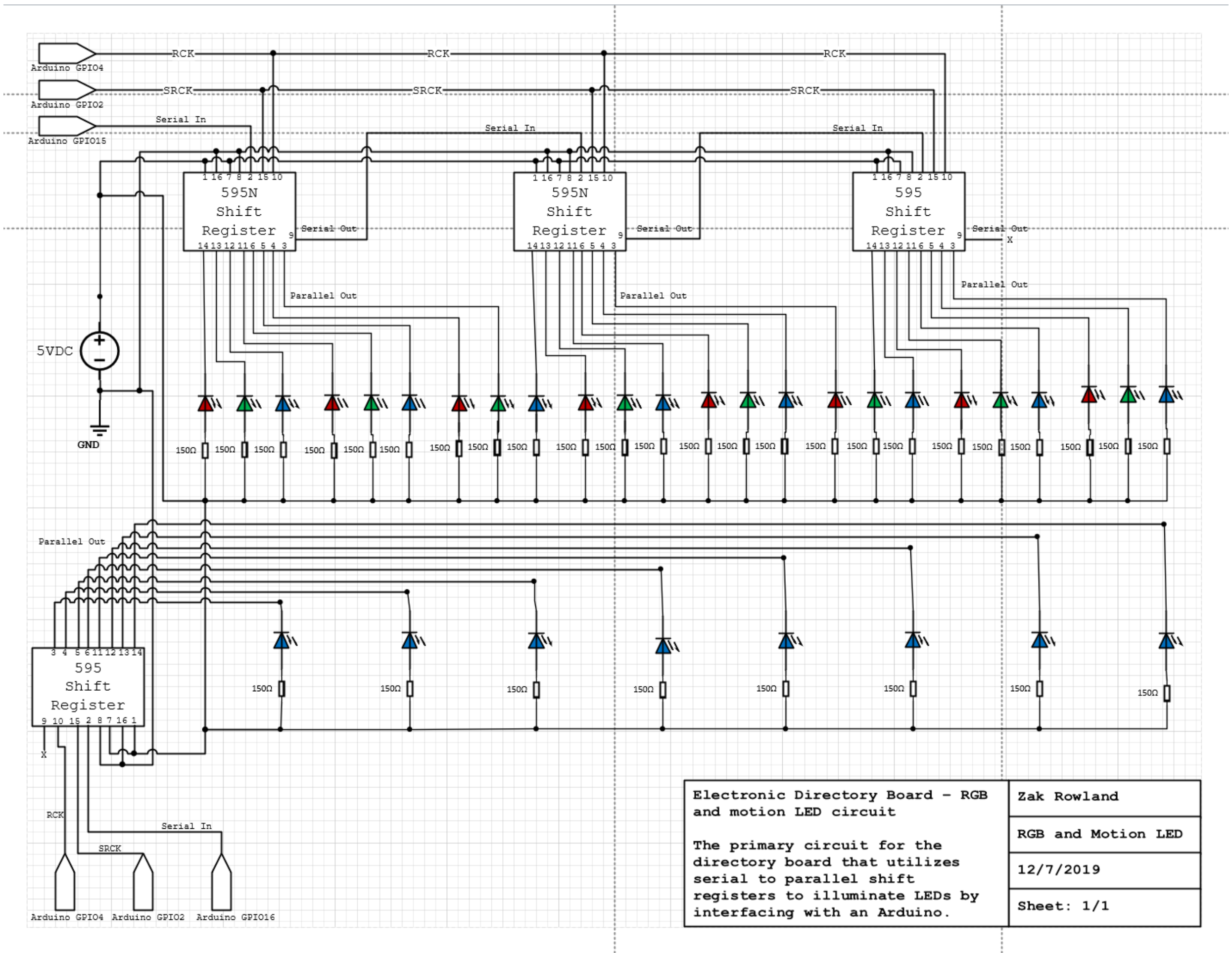

Figure 5. Schematic of the PIR's Interfacing with the Shift Register

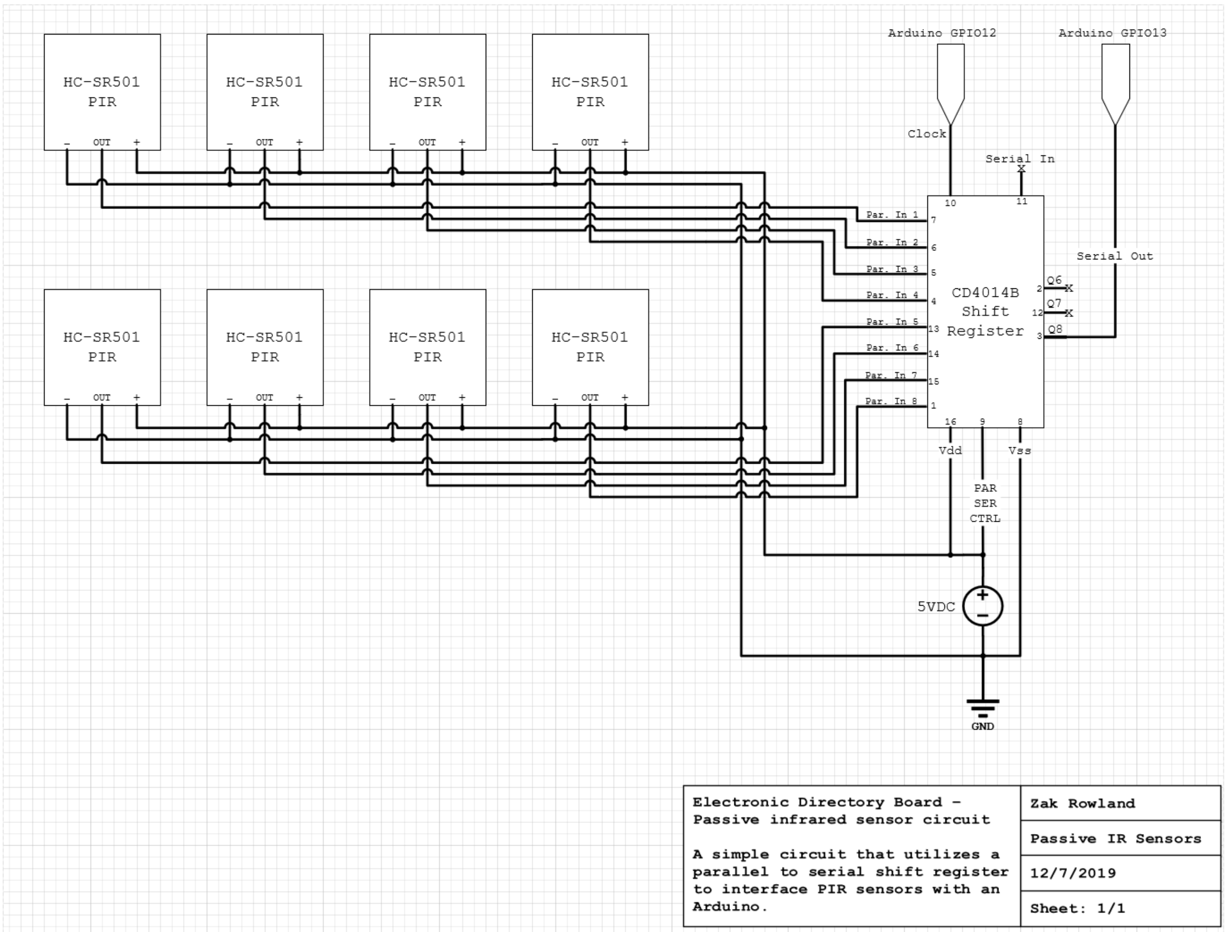

Figure 6. System Block Diagram

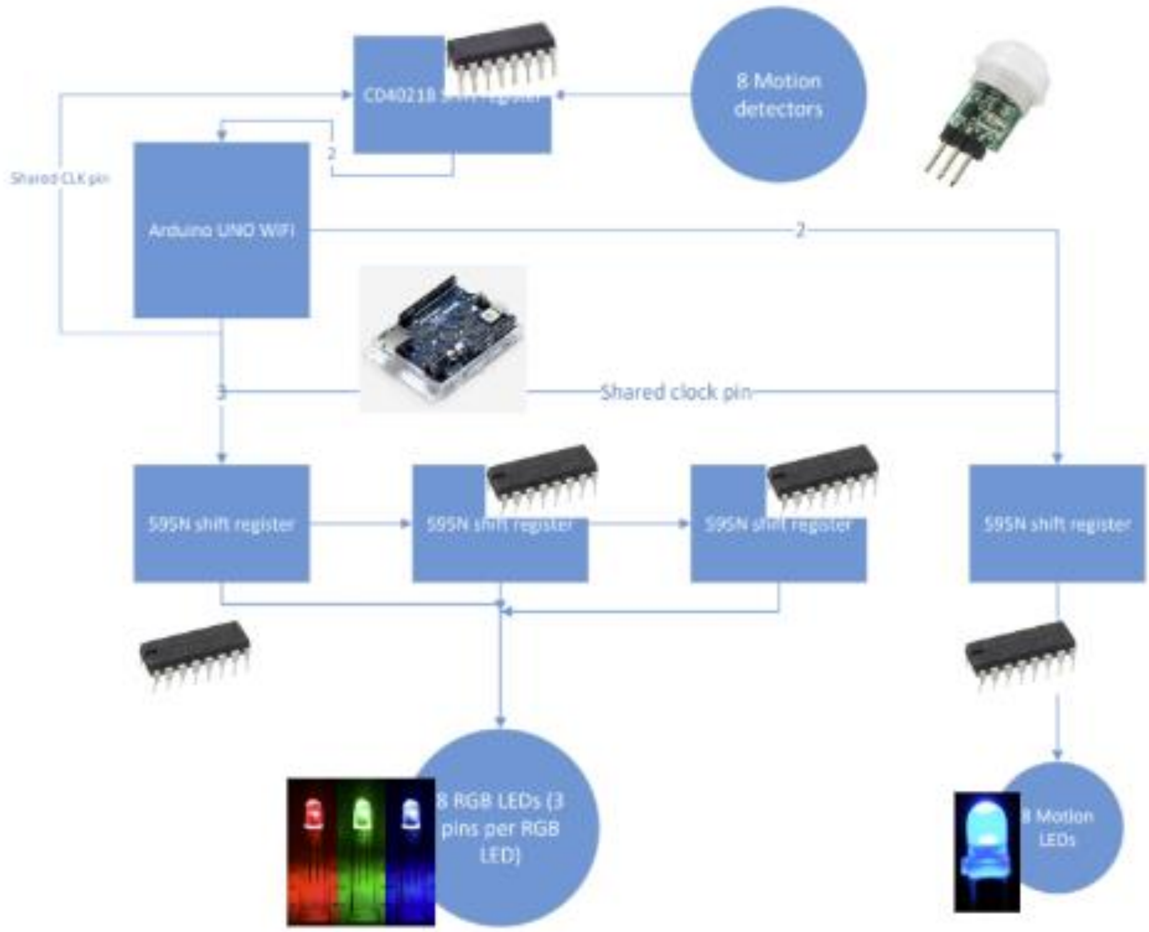

Figure 7. Wi-Fi Diagram

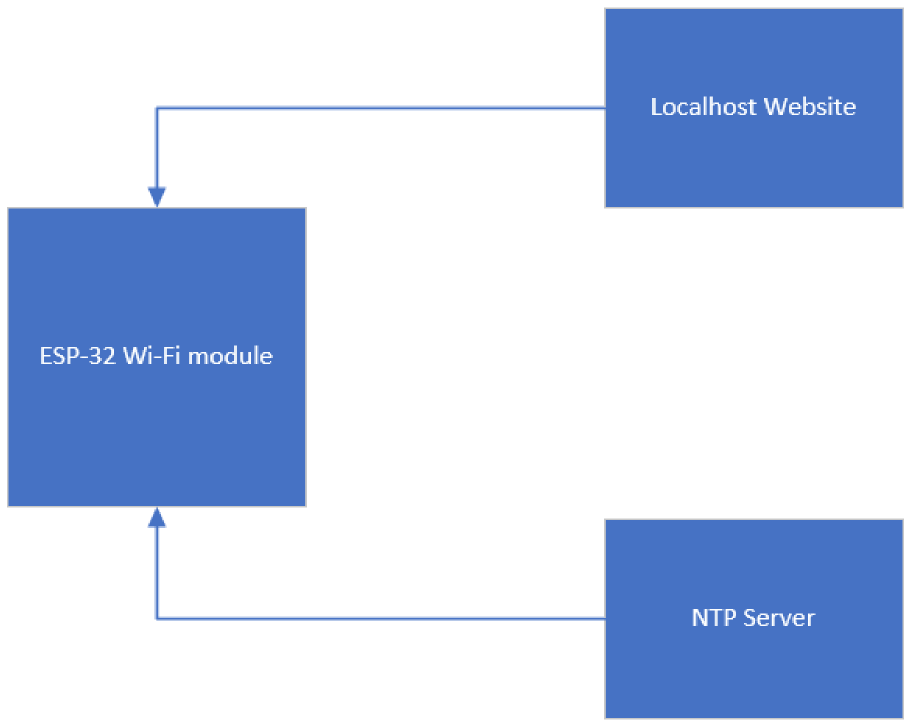

Figure 8. Localhost Wi-Fi Diagram