Blind Me With SciEEnce
EJ Hinlo, Caitlin Gruis, Chris Ravasio
First Design Review

System Block Diagram

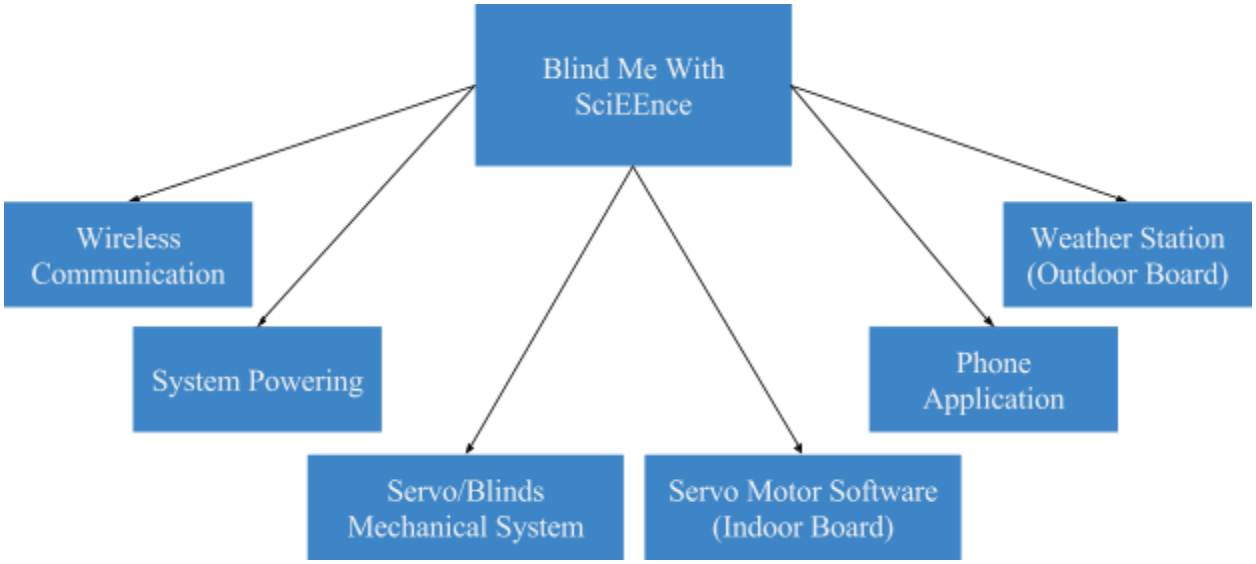

Figure 1. System Block Diagram with Subsystems

Subsystem Communication Diagram

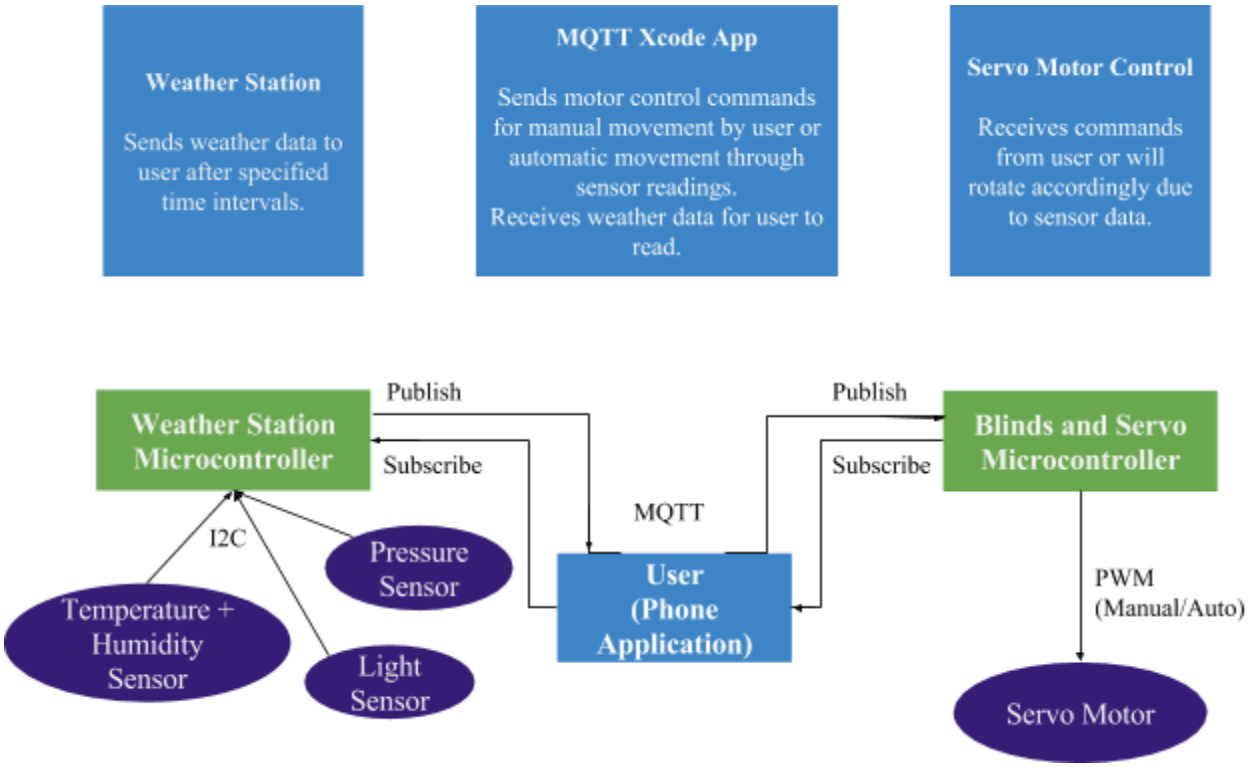

Figure 2. Subsystem Communication Diagram

Final Overall System Requirements

The final overall overall system requirements include the following:

- A blinds system that can be operated both automatically and manually. The automatic mode involves responding to changes in outside light levels and opening or closing accordingly. The manual mode involves sending user-specified angles to set the exact position of the blinds through a servo motor.
    - If time allows, there is the potential to add a timer feature for the user that falls under this overall requirement. The blinds would be able to adjust based on user-specified times.
- A mechanical system consisting of a servo motor that provides sufficient torque to rotate the blinds, fits appropriately with the blinds, and doesn't require excessive power consumption.

- A weather station that provides the user with real-time weather data, including temperature, barometric pressure, light, and humidity. The station will be placed outside and require an appropriate protective covering. To avoid excessive power use and run off a battery, the weather station needs to operate at a low power consumption and send only periodic updates to the user.
- An energy efficient powering system. Parts for the outside board need to operate at a low voltage and current consumption conducive to a LiPo battery and solar charging. The indoor board needs to be able to operate at various voltages due to the requirements of the ESP module and the servo motor.
- A wireless interface based on MQTT protocol through the ESP module that provides smooth communication between devices. Devices will communicate with each other by subscribing to preset topics and publishing necessary information for all systems to interact effectively.
- A user interface in the form of a mobile phone application that allows the user to access all other systems. The phone application will use the publish/subscribe features of the MQTT protocol to communicate between different subsystems.

Subsystem Requirements

1. **Wireless Communication**
    - The underlying interface of the wireless communication subsystem needs to be the MQTT protocol. This is due to its publish/subscribe features and ease of use with the ESP8266 and Arduino IDE.
    - Each set of blinds or weather stations will be hardcoded to subscribe to specific topics in order to differentiate between the various blinds in a user's home. This will be done within the ESP code.
    - It is also necessary for the MQTT protocol to be compatible with Xcode in order to create a phone application.
    - Amazon Web Services will be used as a cloud-based MQTT server. This server acts as a mediator between the ESP clients and the phone application. This also eliminates the need for a physical server in the form of a Raspberry Pi.

2. **Servo and Blinds Mechanical System**
    - A standard size servo is necessary to not only fit in the blinds mount, but also provide enough torque to turn the entire blinds contraption.
    - Mounting brackets or adhesive are needed to keep the servo in place as it performs rotational movements.
    - A connection, in the form a shaft coupler, is required to connect both to the hex rod of the blinds and the spline of the servo.

- A wooden frame is necessary to accommodate the blinds for demonstrational purposes.
- Reasonably priced and accessible blinds are needed to stay within the project budget.

3. **Servo Motor Software**
    - A standard, high torque, non-continuous rotation servo is necessary for setting the exact position of the blinds through a PWM output.
    - In order to avoid breaking/over-rotating the servo and blinds, testing is needed to properly adapt the position of the servo to the various positions of the blinds.
    - The software must be able to accept information from both the user and the weather station to change the position of the blinds.
    - Being able to control the speed of the servo is a must to prevent the blinds from breaking and over-rotating.

4. **Weather Station**
    - To obtain accurate weather data, sensors that measure temperature, barometric pressure, luminosity, and humidity are needed.
    - Each sensor should have I2C capability due to the limited number of I/O pins on the ESP microcontroller. I2C only requires two signal lines and multiple devices can be connected on the same pins.
    - Sensors should operate at low currents and 3V - 3.6V.
    - To prevent damage from outside elements, the final board should be encased in a protective covering that allows for accurate sensor readings.
    - The sensors should be reasonably priced and easily accessible when taking into account the budget of the project and the price of the final product.

5. **System Powering/Charging**
    - Due to the nature of the outdoor board and the infeasibility of running wires through a window, the outdoor board needs to function independently off of a single 3.7V, 850mAh, 3.1Wh LiPo battery, a 2W solar panel, and a charging circuit that connects the two and protects the battery. These values were determined in order to ensure that the battery would not discharge faster than it is being charged.
    - To avoid the difficulties of turning WiFi on and off repeatedly, it is a requirement that the indoor board be constantly powered. Therefore, the board will receive its power from a 5V wall adapter with a microUSB connection.
    - A DC/DC converter needs to be implemented to limit the amount of voltage across the ESP module for the indoor board.

6. Phone Application

- The phone application needs to be developed in the latest version of Xcode and programmed in Swift in order to run on iPhone devices.
- A user-friendly interface is required for intuitive control of the blinds and easily readable information from the weather station.
- On the blinds side, different modes need to be created for manual control, automatic control, and potentially timer control.
- On the weather station side, information about the current temperature, humidity, barometric pressure, and luminosity needs to be displayed to the user. Potentially, history of this data will be displayed to the user as well.
- New blinds and weather stations need to have a way to be registered and added to the application.
- The code behind the app needs to be compatible with the use of the MQTT protocol.