Blind Me With SciEEnce

High Level Design

Caitlin Gruis
Enrick Hinlo
Christopher Ravasio

Table of Contents

1. Introduction

Sometimes, one of the most difficult parts of the day is getting out of bed in the morning. Our senior design project is a “morning companion” that allows the user to have a better experience waking up every day. It combines several unique features, including automated blinds that open at a specific time specified by the user, as well as a weather reading taken from directly outside the user’s window. This weather reading is sent to the user’s phone, alerting them in an app to the current temperature, humidity, and luminosity (cloudy/sunny). It also gives them a suggestion of what to wear based on this information. Additional functionality we may add includes a speaker that would play a predetermined sound to aid in waking the user up instead of having them set a phone or clock alarm. We could also run the microcontroller primarily via solar power instead of a battery in order to make the product more energy efficient; the user would never have to worry about changing the battery. Throughout the day the blinds will act as a smart automated system, taking in relevant information regarding light and temperature and subsequently opening or closing the blinds accordingly. For example, if the inside of the house reaches a certain temperature threshold the blinds will close. This action should keep the house cool more efficiently, cutting down on energy costs. Likewise, opening the blinds at sunrise lets the sun help warm the house in the morning.

2. Problem Statement and Proposed Solution

Problem Statement:

Waking up, particularly on a weekday morning to get ready for school or work, is a difficult task that many people do not enjoy. The room is dark with the blinds closed, making it difficult to roll out of bed and begin the day. From the moment someone wakes up, they are immediately plagued with questions such as: What is the weather like outside? What am I going to wear today? All of these combined issues make for a hectic and unpleasant morning routine that can start a person's day off on the wrong note.

Additionally, people spend time in the morning and evening opening and closing the blinds in their house. Furthermore, when people are away from the house for long periods of time, such as on vacation, leaving the blinds permanently in either an open or closed position is an indicator to burglars that no one is home. Leaving blinds open during the day allows the sun to heat a room past the thermostat set temperature, resulting in higher energy cooling costs. Conversely, leaving blinds closed early in the morning misses out on a chance to more efficiently heat the home at the beginning of the day. Also, many modern homes have windows in high or hard to reach places, making it difficult to adjust blinds or impractical to have them at all.

Problem Solution:

This project seeks to make the morning experience more enjoyable and less stressful by quickly answering some of these daily questions for the user, as well as opening up the blinds to illuminate the room to prompt the person to get out of bed and wake them to natural sunlight. The blinds will operate via rotational motor and will open/close and/or raise/lower at a specific time set by the user. This wake up time may be accompanied by a noise or phone alarm. Additionally, the weather station will take various data readings at that time, analyze the corresponding data, and notify the user of the weather for the day and what items of clothing are appropriate to wear. This combined app + automated blinds + weather station interface will make for a much more pleasant and prepared morning. The weather station will be equipped with various sensors needed to gather important weather data (light, temperature, humidity, etc.) The app will allow the user to set various parameters such as the time for morning wake-up, and it will provide the user with controls for the blinds. We plan on having a wireless interface for everything to communicate with ease. Lastly, to solve the other blinds related problems, the blinds will automatically adjust throughout the day based on sensor readings inside and outside the house. With this setup, problems with forgetting to adjust blinds, having windows in hard to reach places, adjusting blinds while away on a trip, or using the blinds to help cool or heat the house are solved.

3. System Requirements

- **What is the nature of the required embedded intelligence?**

Our embedded intelligence is going to be fairly simple. The ESPS microcontroller is probably going to be the most advanced piece of hardware in our project, and it will only have to make fairly simple decisions. The software will be making choices like whether to open or close blinds based on whether a temperature or light threshold is currently met, or if a command has told it to. This should not require any advanced intelligence on the chip outside of the ESPS.

Our system will have to transfer data from the sensors on our chip to the app on a person’s phone. This will require an analog to digital converter as well as wi-fi. Data will also need to be transmitted from the phone app to the microchip. Since we will likely also have two separate boards, one inside and one outside the window, we will need to use wi-fi to transfer relevant data between the two boards to whichever board the “main” decision making ESP is on. Further, we will likely require an ADC on both boards as well since we have a temperature sensor inside as well.

· **How is the device powered? If it runs on batteries, what kinds of batteries are used, how long should the system be able to run on the batteries, etc?**

The two boards will be powered by solar rechargeable batteries. The batteries need to provide enough voltage to power the motor. Otherwise, the batteries only need to maintain the system while it is in sleep mode and provide slightly more power when the system wakes up to check sensors and command requests.Ideally, the batteries will never be drained fully because they are able to recharge while the system is in sleep mode. If we are unable to get the motor to run off solar batteries, we will have to use a wall wart to power the indoors system.

· **There are lots of requirements related to wireless interfaces. How many devices need to be supported? What range is required?**

There will be three devices interacting with each other on a home wireless network for our project. We will have two ESP chips, one on each of our boards, as well as a smartphone app. The range for the ESP chips will not need to be very great, as they are stationary and should always be connected to the home’s wi-fi. The smartphone app may need much greater range if we decide to let the user interact with their blinds from anywhere using the app.

· **What are the user interfaces?**

The only user interface is the smartphone app. All direct interaction with our product is handled through the app, and the user never has to handle the hardware outside of installation.

- **How is the system installed and used?**

The indoors board and motor component will be mounted inside the blinds mounting area where the hex rod is located. The outdoors board will be secured outside the window, either on the window sill or the top of the window. The outdoors board will also be protected from the elements in some manner. The system is used through settings and commands from the smart app, as well as on-board decisions predicated by the environment.

- **If you project involves voltages and or currents that may be dangerous, what are safety requirements associated with your system.**

All of our systems will be fairly low voltage and low power, so we shouldn’t have any serious safety requirements.

- **What are the mechanical requirements, such as weight, size, etc.**

The only major mechanical requirement is that our servo motor has enough torque to open and close our blinds. If we want to raise and lower the blinds as well, this would require even more torque from our motor. We have already demonstrated that our blinds open and close with our motor, so the only thing we need to worry about is raising and lowering the blinds. The other physical requirements are making our indoors board fit inside the blinds mount and the outdoors board adequately protected and mounted.

4. System Block Diagram

4.1 Overall System

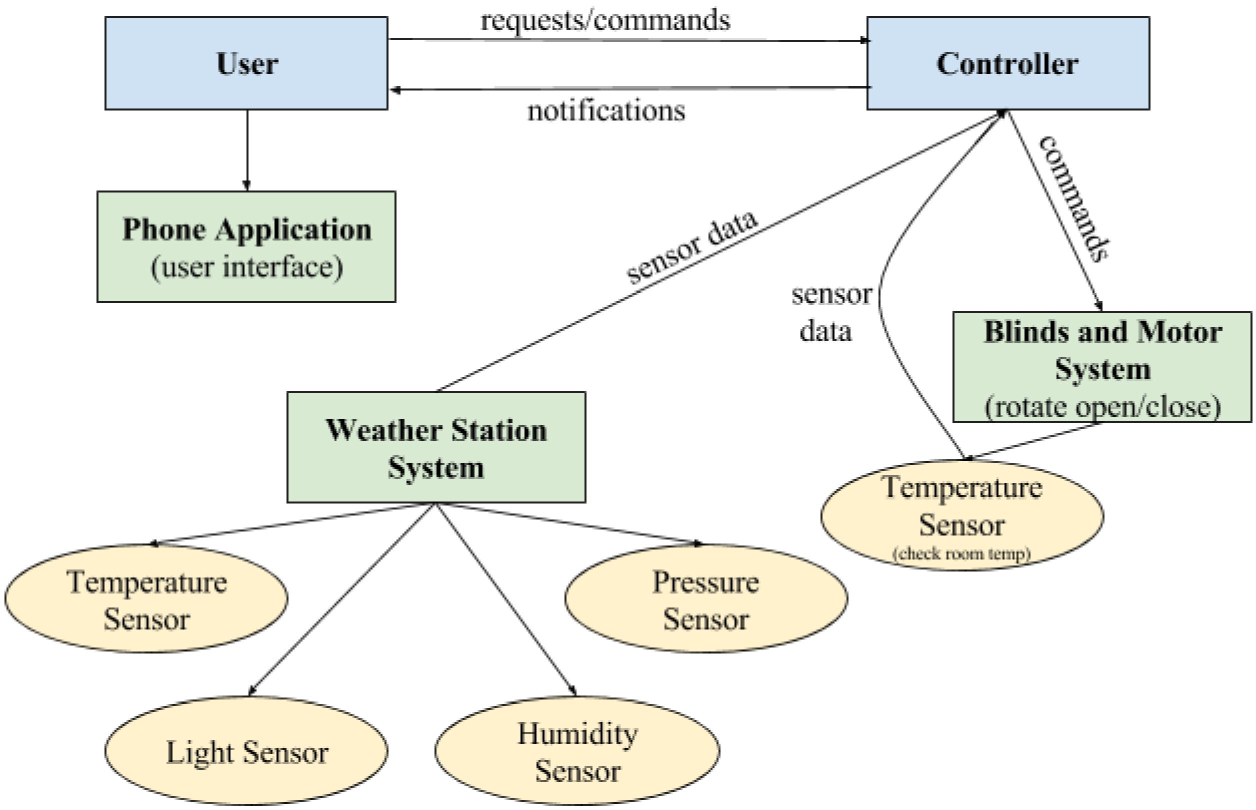

Figure 1. Overall System Diagram

The overall system involves a user and a controller. The controller includes both a weather system station and a blinds and motor system. The weather station system is located outside the house, and takes current weather data using a temperature, light, humidity and pressure sensors. This information is sent back through to the phone application that the user is interacting with to inform the user of the current weather. The blinds and motor system is located inside the house, and autonomously rotates the blinds based on the time of day and user preferences. There is a temperature sensor included in this system to check the inside temperature, and adjust the blinds based on whether or not it is hot or cold in the house. On the user side of the system, there is the phone application. The phone application is receiving information from the weather station about the current weather, and it will also allow the user to manually control the rotation of the blinds.

4.2 Subsystem and Interface Requirements

Phone Application

The software side of this subsystem includes the following:

- Xcode 7 - This is the programming environment that we would use to create an iOS based phone application.
- iOS 9 - The is the current version of iOS that works with the most recent Xcode. This operating system can be run on 6th generation and later Apple devices.
- Parse.com or some sort of backend service - This is where the user's preferences will be saved about when they want the blinds to open and close throughout the day, and when they want to be woken up in the morning.

The hardware components of this subsystem includes the following:

- 6th generation Apple devices (iPhone 5/5s and later). These systems support iOS 9 which is a requirement for being able to program on the latest version of Xcode. The only differences between Apple devices with iOS 9 would be screen size.

Blinds and Motor System

The software side of this subsystem includes the following:

- Pulse width modulation (PWM) would be needed to control the motor connected to the blinds. This is a type of digital signal that is used in a variety of applications including sophisticated control circuitry. This type of signal processing is important for servo motors because with it you can set the exact position.
- Wireless communication - The ESP8266 part will be used for this. It is described in more detail in Section 5, but data to convey the current blinds position will be sent from the system to the app interface, and the app interface will send data to the system to set the desired new position.
- Code to control blinds based on temperature and user preferences - This will be programmed into the microcontroller. If the temperature sensor reads above a certain value, the blinds will kept open, but if it drops below that, the blinds will close. The user can also set the blinds to open and close at a certain time of day.

The hardware components of this subsystem includes the following:

- The main mechanical part of the project relies on the use of a servo motor to control the hex rod of the blinds.
- A temperature sensor is necessary for detecting the temperature of the room and sending this data through the wireless interface to the controller to adjust the blinds accordingly. A hot room would close the blinds, and a cold room would open the blinds.
- 2x4 wooden pieces will serve as a frame for the window, and the blinds will be mounted onto this.

- A circuit board will be used to support wireless communication. It will include pins for motor as well as headers for programming. This will be placed on the inside of the window.
- A rechargeable battery with a solar power charger will also be included.

Weather Station

The software side of this subsystem includes the following:

- Code is necessary to be able to turn on the various sensors on the board and keep track of the data coming from each. Further code is also necessary to setup wireless communication with the phone and the blinds in terms of the light sensor.
- Xcode might be necessary to format the data taken from the weather station in a friendly user interface. Some sort of database server or backend service may be needed to save weather data for trend analysis or to save user preferences based on light detection.

The hardware components of this subsystem includes the following:

- Circuit board with temperature sensor, light sensor, humidity sensor, and pressure sensor. This will most likely be our main board, and will be placed on the outside of the window. This will be connected to the microcontroller which has the necessary pins, inputs, outputs, and power pins for proper functionality.
- Some form of enclosure will be needed to protect the weather station from inclement weather and/or moisture.

- A rechargeable battery with a solar power charger will also be used to power the weather station and provide enough power to activate sensors and power the microcontroller that is also connected.

4.3 Future Enhancement Requirements

Some of the future enhancements to be considered in this project are:

- Adding the ability to also raise and lower the blinds (in addition to the rotation of the blinds included in the initial product)
- Using the position of the sun to adjust the blinds accordingly in order to block a certain amount of sunlight (blind position would be in between fully open and closed)
- Adding rain and wind sensors to the weather station
- Setting your phone alarm through the app or use a buzzer on the microcontroller

5. WiFi considerations

5.1 Connection Considerations

A single WiFi connection to a secure or unsecure network will be used. There is no reason that this product would need to be connected to multiple networks. We are assuming that the device will be used in the home, and thus be able to connect to a home WiFi network. Once it is connected to this network, it should not need to change networks at any point in time. One other thing that we will need to take into consideration with this connection is that it would be ideal to be able to manage the blinds position and see the weather information from anywhere.

For example, if the user is at work, they would still be able to open and close their blinds and see what the temperature is outside of their house.

5.2 Data Flow

The following figure shows the data flow between each of the systems involved in this project:

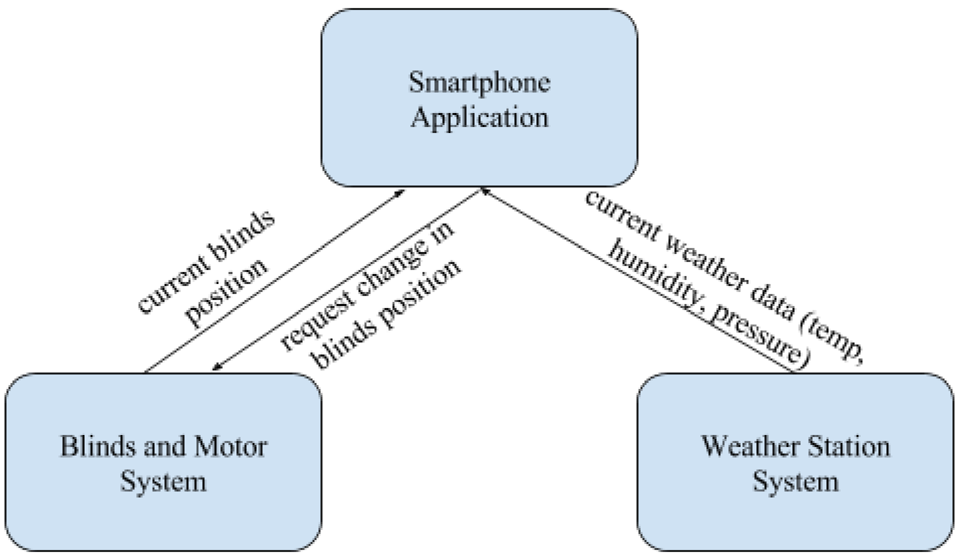

Figure 2. Wireless Data Flow Diagram

The nature of the information exchanged would be simple data. The weather station system would be sending sensor data from a temperature sensor, humidity sensor, and pressure sensor to inform the user of the smartphone application of the current weather. The blinds and motor system would be sending data to the smartphone application about the current position of the servo motor (presented to the user as the current blinds position), and the smartphone application would be able to send a request to the blinds and motor system to change the current position of the motor.

On the weather station side of things, data would only need to be sent around once every hour, because the weather does not tend to change very quickly. There could also be an option for the user to send a request to the weather station system to send data more frequently, if they chose to. Information about the position of the blinds would need to be sent to the smartphone app every time the position changed, and then the request to change the position should be sent to the smartphone whenever the user wants it to change.

At this point, we are unsure whether or not it will be necessary for a server and database to be involved. We think that this may be required if we want to do something such as save previous weather data that has been taken (so that the user can see past trends), or to show what the blinds positions have been throughout the day.

5.3 ESP8266

Because of the nature of our data flow, we believe that a separate microcontroller is not required, and that the ESP8266 can perform all of the required functions. At least for initial development, we are thinking of using the SparkFun ESP8266 Thing model for this project (available at this link: https://www.sparkfun.com/products/13231?gclid=CJTC2PWDy8kCFQyFaQodBzgC1Q), because it nicely breaks out all of the modules pins, there are excellent schematics and documentation available for this model, and it looks like it makes it fairly easy to interface it with sensors and motors.

If possible, it seems the best programming environment for this would be in the Arduino IDE. The ESP8266 was created with this IDE in mind, and there is a lot of sample code and instruction available to help guide us in using this part if we choose this programming environment. Using the Arduino IDE would also make interfacing the sensors a less of a complicated process.

In creating our smartphone app, we will also need to use the Xcode programming environment to communicate with the ESP8266. This is our only choice in a programming environment if our goal is to create an application that will run on an iPhone. We could also consider using the Android Studio programming environment to extend our project to communicate with Android phones as well.

5.4 Protocol for Data Exchange

Our system will be using MQTT as the protocol for data exchange. MQTT looks to be an excellent protocol for Internet of Things projects because it requires a low usage of power, minimized data packets, and it is able to efficiently distribute information to one or more receivers. These are all desirable characteristics for our project. This protocol works as a publisher and subscriber type of messaging transport. So, for example, the phone application would be a subscriber to both the weather station system and the blinds and motor system to receive the data that those devices are publishing, and the blinds and motor system would be a subscriber to the smartphone application so that it could be aware of the user sending it requests to change the blinds position.

6. High Level Design Decisions

Each subsystem can be analyzed and described by the high level decisions that characterize and correspond to each of their requirements. Our main subsystems are the mechanical blinds component and accompanying microchip/board, the weather station, and our smartphone application.

The blinds system will consist of a mounted, high torque servo motor that rotates the hex rod of blinds, an ESP8266 chip to communicate with the motor system as well as make software informed decisions to open or close the blinds, and an indoor temperature sensor. The indoor temperature sensor will be used to help intelligently heat or cool the home via sunlight. The ESP8266 chip will make decisions to open or close blinds based on sunlight or temperature, as well as from user requests through the app. The system will be powered either through solar rechargeable batteries and a solar cell, or via a wall wart plug-in. Hopefully we can get the solar power to work, because this would both save energy and be more aesthetically pleasing and user friendly.

The weather station system will include a temperature sensor, humidity sensor, barometric pressure sensor, and light sensor. These are the best sensors that we know of to accurately and simply take current weather data. There are also rain and wind sensors available for purchase, but they are bulky and more complex to work with. Therefore, they will be additional features to our project if time allows. This system will be powered through a rechargeable battery and a solar cell. This type of powering allows for the least inconvenience for the user, as they will probably only have to change the batteries every couple of years.

Our final subsystem includes the phone application we will develop that will be accessible on a 6th+ generation Apple device such as iPhone 5, 5s, or 6. Xcode 7 will work as long as the device has iOS 9. This subsystem will allow the user to interact with the blinds via the ESP8266 and a corresponding MQTT publisher/subscriber server that grabs the data sent from the motorized blinds system and the weather station system. The application will be user friendly and will potentially save user-preferences and settings to some sort of backend service such as Parse.com. Specific times and other settings can be applied to the overall system for automation or manual control by the user. The application will also display pertinent weather information for the day and has the option to include what to wear for the day.

7. Open Questions

- To what extent do we need a backend service/database server? We would only need this to save user preferences for when the blinds should open or close and when to wake up every morning.
- How can we obtain a balance between protecting our weather station from the outside conditions such as wind and rain, and still getting an accurate weather reading?
- Where outside will we be mounting the weather station, and how will we be mounting it?
- Is it necessary for us to build our own blinds with its own plastic frame, hex rod, and blind rotators?
- How feasible is it to also raise and lower the blinds if we finish the main requirements of our project?

- Our plan is to use a rechargeable battery with a solar charger for the weather station, but should we use the same setup for the indoor blinds motor system or should we just plug that into an outlet? Continuing with this question, what are the best components for a rechargeable battery + solar charger backup setup?
- How, if necessary, will we detect blind position?
- Will we change blind position based on temperature, sunlight, or both? What is our balance of changing the blinds between temperature and sunlight?
- Will we let the user interact with the blinds from any location, e.g. while they are at work or otherwise away from the house?
- Will the sensors be soldered onto the board or attached in some other way?

8. Major Component Costs

The following parts are for the blinds system of the project:

- Faux wood blinds ~ $15.00 - $50.00. The price of blinds depends on how high quality we want our project to be. There's also the option of constructing our own set of blinds (faux wood panels, plastic frame, hex rod, blind rotators, etc.). The advantage to this is creating a frame to not only encompass the tilt gears, cradle, and hex rod, but also the circuit board and motor.

  http://www.blindsparts.com/wood-blinds/
- 2x4 wooden pieces ~ $5.00 - $10.00. These wooden blocks will be used to build a window frame for our blinds. *Bought at Home Depot.

- Generic high torque servo motor ~ $12.95. This will be the main mechanical system of our blinds. It provides enough torque with low power to rotate the blinds in conjunction with the hex rod.

  “Here is a powerful, low-cost, reliable servo for all your mechatronic needs. This servo is able to take in 6 volts and deliver 83.47 oz-in. of maximum torque at 0.16 sec/60°”

  https://www.sparkfun.com/products/11965
- Servo motor bracket mount ~ $10.00 - $15.00. We need to buy an appropriate servo mount to fit the servo to the hex rod holder while also keeping the servo stable.

  http://www.robotshop.com/en/lynxmotion-aluminum-multi-purpose-servo.html
- Temperature sensor ~ See temperature sensor under weather station component.
- ESP8266 ~ $15.95. “The SparkFun ESP8266 Thing is a breakout and development board for the ESP8266 WiFi SoC – a leading platform for Internet of Things (IoT) or WiFi-related projects. The Thing is low-cost and easy to use, and Arduino IDE integration can be achieved in just a few steps.”

  https://www.sparkfun.com/products/13231?gclid=CJTC2PWDy8kCFQyFaQodBzgC1Q
- PCB ~ $50.00+. This will most likely be our main board. It must contain our main microcontroller and pins for servo, PWM signal processing, sensor, and ESP8266 connection. This board will be the bigger of the two needed for our project due to the requirements of this system.

  http://www.4pcb.com/index.html
- Rechargeable Battery* - To be determined. See Open Questions
- Solar Charger* - To be determined. See Open Questions

The following parts are for the weather station system of the project:

- PCB ~ $50.00. This will be the smaller of the two board. It only needs the microcontroller, power, and pins for all the sensor connections.
  http://www.4pcb.com/index.html
- ESP8266 ~ $15.95. “The SparkFun ESP8266 Thing is a breakout and development board for the ESP8266 WiFi SoC – a leading platform for Internet of Things (IoT) or WiFi-related projects. The Thing is low-cost and easy to use, and Arduino IDE integration can be achieved in just a few steps.”
  https://www.sparkfun.com/products/13231?gclid=CJTC2PWDy8kCFQyFaQodBzgC1Q
- Temperature Sensor ~ $1.50 - $7.50. This part would ideally be soldered onto the board. I included links to both cheap, single components to a much larger board with more accuracy, intelligence, and pins. Datasheets are included for our own construction.
  https://www.sparkfun.com/products/10988
  https://www.adafruit.com/products/2635
- Light Sensor ~ $1.50 - $5.95. This part would ideally be soldered onto the board. I included links to both cheap, single components to a much larger board with more accuracy, intelligence, and pins. Datasheets are included for our own construction
  https://www.sparkfun.com/products/9088
  https://www.adafruit.com/products/439
- Humidity Sensor ~ $7.25 - $19.95. This part would ideally be soldered onto the board. I included links to various types of boards that have humidity sensors. Datasheets and schematics are included for our own construction.

https://www.adafruit.com/products/2635

https://www.sparkfun.com/products/9569

https://www.sparkfun.com/products/11295

- Barometric Pressure Sensor ~ $10.00 - $60.00. This part would ideally be soldered onto the board. I included links to search pages of boards that have fairly accurate pressure sensors. Datasheets and schematics are included for our own construction.

  https://www.adafruit.com/search?q=pressure&b=1

  https://www.sparkfun.com/search/results?term=pressure

- Rain Sensor* ~ $4.00. This is a potential future addition for our weather station. However, no schematics or datasheets are found.

  http://www.dx.com/p/raindrops-sensor-module-blue-black-199859#.VmmpMvkrKhf

- Wind Sensor* ~ $17.00. This is a potential future addition for our weather station. Datasheets and schematics are included for our own construction.

  https://moderndevice.com/product/wind-sensor/

- Rechargeable Battery* - To be determined. See Open Questions.
- Solar Charger* - To be determined. See Open Questions.

** - Parts that are potential future additions or have yet to be determined.*

9. Conclusions

References

http://www.instructables.com/id/Using-the-ESP8266-module/

http://neilkolban.com/tech/esp8266/

http://fab.cba.mit.edu/classes/863.13/people/ermal/mas.863/final-project-ambilinds-sunlight-driven-automated-window-blinds/

http://mqtt.org/

https://www.sparkfun.com/products/13231?gclid=CJTC2PWDy8kCFQyFaQodBzgC1Q

https://homeawesomation.wordpress.com/2013/02/26/automated-window-blinds-with-arduino/