A.I.R.E – Autonomous Intelligent Roof Enclosure

Lauren Miller, Sarah Riseden, Christopher Smith, Ronald Acevedo

Sponsor: OpenAire-Retractable Roofs and Skylights

Dept. of Electrical and Computer Engineering

University of Central Florida, Orlando, Florida

***Abstract* – This paper details the A.I.R.E. autonomous roofing system, which appears as a retractable glass roof operating on a wide variety of weather sensors. These sensors allow for the roof to be considered a smart systems, which can react to weather as well as predict incoming weather. Accompanied by an iOS app, this system allows for user to interface communication as well as customization by the user. The app will be readable and allow the user to implement their own settings for the roof operation, provide real-time weather statistics as read by the sensors, as well as connect to a local weather API for further weather updates and communication with the implemented sensors.**

***Index Terms* – Autonomous, Roof, iOS Application, Bluetooth, Solar Panel, Battery, Weather.**

I. Introduction

As the idea of a smart item, a.k.a. the Internet of Things technology grows in popularity, it has begun to appear in everything. The most demand being found in the smart home. In order to take the automation of a building even further, the next step would be an automated roof. In order to create a comfortable experience at any location, the roof would be retractable and glass, bringing the outside in. This system would be appropriate over facilities that can only operate comfortably in specific weather conditions but would like to continue functioning even when Mother Nature isn't cooperating. In the current market, roof systems like this exist, as created by our sponsor OpenAire. However, this collaboration will allow for further research into the system, by testing any updates and modifications before creating something full scale. This can aide in the company's growth, in order to go above and beyond any other competitors on the market.

In an attempt to solve the issue of weather restrictions, we propose a fully automated retractable glass roof. The automation would come from various sensors to provide the most comfortable outdoor/indoor experience. These sensors would be used primarily for weather predictions to detect the following characteristics of weather: rain, wind speed, air pressure, temperature, humidity, and cloud coverage. The roof will act preemptively according to the outside weather conditions and will be able to detect if unwanted weather is on its way, not just when it arrives.

This system will also come with a locked app available on iPhone, that only allows owners of the roof access. This allows the user to control the system via app and set their own thresholds for operation of the roof and lighting depending on the setting the roof encompasses. With this app, the user can build their own packages of settings and create a schedule on the roofs operation to fit business hours, daylight hours, etc. Along with providing full control of the system, the app will include the statistics read in from the weather sensors, allowing the user to make educated decisions when creating their own roof settings. The app will mainly connect via Wi-Fi as well as having Bluetooth capabilities for backup. In case of emergencies such as a power outage, Wi-Fi outage, or sensor failure, the roof will be connected to an emergency power source, and building will have a main switch inside which will override all app settings and sensor data, and close the roof.

II. System Design

The A.I.R.E. enclosure contains multiple different systems all working together to control the roof's operation. There are many sensors collecting and sending data to the development board in unison. The development board then turns this data into readable information in real time, and sends it to the app for user interaction. The development board makes the decision about whether the roof should be open or closed according to the analyzed sensor data. Powering all of this is a solar panel and series connected lithium ion batteries, all regulated by a battery management and protection PCB and various voltage converters.

III. System Components

The following sections detail all system components including hardware and software components. This will include important decision details and an overall view of what systems will be used in this project.

A. Microcontroller Unit

For the final project development, the microcontroller chosen is the ATMega1280. A microcontroller with enough input/output pins was required as there are many components connected sending and receiving data. The ATMega 1280 has 54 GPIO pins, a 16-channel, 10-bit ADC converter, and 128kB of flash memory. This was deemed to be the optimal microcontroller as it allows for storage and execution of sophisticated and complex algorithms with ease.

Unfortunately, due to the impacts of COVID-19, the original design and use of the ATMega1280 was not able to be implemented in this project. To replace the initial design, it was decided to use the Bluno Mega 1280, which is a development board with an ATMega 1280 chip on it. All specifications are the same as the ones stated for the ATMega chip with the exception of a higher operating voltage range, which has proven to add some complications to the programming of this board. This board also has Bluetooth capabilities with a fair amount of documentation to assist in the setup of this board and its connections.

B. Sensor Module

The sensor module contains five sensors, all used to collect current weather information, and aid in the prediction of inclement weather. These include sensors for: humidity, temperature, rain, light, and UV.

The humidity sensor of choice became the SHT-85 from Sensirion. After testing other humidity sensors with similar functionalities, this one fit our system the best. This sensor also includes a temperature sensing function, which be used as a backup to the temperature sensor that is also implemented in the sensor module. The SHT-85 also has self-calibration capabilities if it is in conditions that are outside of normal. This function allows for accurate measurements without much maintenance or user interference.

Along with the SHT-85, an additional temperature sensor was implemented for better weather analysis. The B57863S used in our system is a small NTC bead thermistor. The thermistor is calibrated at a nominal resistance of around 3kOhms and measures temperature by decreasing resistance as temperature rises or increasing resistance as the temperature lowers. The change in resistance creates output voltage changes allowing for a straightforward implementation onto the PCB and into the software.

Finding a rain sensor that would work well with the systems applications, that also has adequate documentation, and wasn't too costly proved to be a challenge. However, rain sensors are one of the most important sensors required for the weather forecasting and controlling the roof's operation. The final decision was the YL-83, which is a resistive collection board sensor. This sensor also comes with an electronic board which includes a potentiometer for sensitivity adjustment, a power LED, a digital output LED, and some stabilizing components.

Finally, a light sensor and a UV sensor were added to the system. The chosen light sensor, OPT101, is to be used mainly for time of day purposes, whereas the chosen UV sensor, VEML6070, is used to detect harmful rays to keep customers inside the enclosure safe. The OPT101 is an integrated circuit photodiode and transimpedance amplifier with built in leakage current error, noise, and gain peak protection. The VEML6070 is also an integrated circuit photodiode with a lowpass filter and I2C supported circuitry.

C. Motor Module

To give the roof movement, a motor must be implemented into the system. For this, the Progressive Automation's -07 linear actuator was chosen. This actuator extends fully when given a +12V DC signal and retracts when given a –12V DC signal. In order to achieve this voltage switch and keep the system smart physical switches cannot be used. Instead, single pole double throw (SPDT) relays have been chosen for this application. These relays are 5V relays and will be connected to the system's microcontroller through a 5V relay board (The voltage necessary to power the microcontroller) [p]. The combination two SPDT relays, allowed us easily to reverse the polarity of electricity going to the actuator motor.

By controlling the polarity going to the motor, we can control the direction of travel for the linear actuator. Figures 2, 3, 4, and 5 serve as demonstration of how the different components are wired up in order to complete this action:

A small wire is used to connect the Normally Open (NO) connection on relay #1 with the with the Normally Open on relay #2. This is for our source +12VDC that will be used for the linear actuator motor. These come from the + 12V Power lines provided from the lithium ion battery discussed in the power section of this experiment.

The same process is done for the Normally Close (NC) pins. A small wire is used to connect the Normally Close (NC) connection on relay #1 with the Normally Close on relay #2. This is for our Ground that will be used for the linear actuator motor.

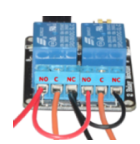


Fig. 1. SPDT configuration step 4

These two wires can effectively be connected to the power line for the A.I.R.E. systems, connecting the first and second wire (red and black in Fig. 1 provided above) to +12VDC and Ground respectively.

With this combination, we can program the microcontroller and trigger the proper relay based on the action that wants to be accomplished. Whether it be to open or close the roof.

Refer to the microcontroller algorithm section of this document in order to understand the logic in the code behind this operation.

D. Solar Power

In order to provide energy saving functions to the system in addition to the current features, a solar panel will be added to the power system. This will take the strain off the battery pack, while providing natural renewable energy, which will save customers money overtime. For this small-scale application, a 12V 5W polycrystalline panel from ECO-WORTHY was chosen. Polycrystalline panels have multiple silicon crystals per cell and are known to have lower energy efficiencies compared to monocrystalline panels with only one crystal per cell. However, the solar panel chosen produces an impressive amount of power for its low cost compared to other solar panels that were considered. It was determined that the 12V supply it is capable of producing was more than enough to power multiple components during peak hours of the day. The solar panel dimensions of 255 mm x 194 mm x 17 mm, make it too big to attach directly to the building, but my creating a solar farm type of setup next to the building, the tilt and placement of the panel can be customized to increase overall efficiency.

E. Battery Management

Because the system design includes a large array of components all requiring varying levels of supply power, a complex power supply and battery management system was designed. for the initial power source, the decision was between lead-acid batteries and lithium-ion batteries. Compared to the other options on the market, the lead-acid battery is very inexpensive. It has been in use for many decades, meaning that the technology used to control this type of battery is very mature and has a heavily-developed support network to provide application notes for dealing with the lead-acid battery. On the other hand, the lead-acid battery has very poor energy density and can only be used for occasional full-discharge cycles.

Compared to lead-acid batteries, lithium-ion batteries have much larger energy density and many available charge/discharges cycles before needing maintenance. In addition to the improved energy density, lithium-ion technology lasts much longer in storage while being capable of discharging more often. While it boasts an impressive list of advantages, the lithium-ion battery requires a battery management system to protect the circuit it is powering as well as itself. Also, the technology is relatively new and, while it does have extensive application notes for dealing with technology, aging of the lithium cells can have a detrimental effect on their performance. Lithium ion batteries are also much more expensive than lead-acid batteries.

Voltage regulation techniques were analyzed to find the most suitable option. Looking at shunt zener, linear, and switching voltage regulators compared to building a custom battery management integrated circuit (BMIC), it was decided that constructing a custom battery management system utilizing the battery management integrated circuit (BMIC) would be the best candidate for its role in managing the charging cycles for the lithium ion battery pack. With its active role in regulating the charge cycles for the battery and preventing over-voltage and overheating, the BMIC is the most powerful candidate for the BMS system while keeping costs and board size low. It will also be capable of handling the amount of power that the solar panel will provide (5 W). For all other power regulation, buck/boost converters will be used to efficiently maintain voltage and ensure that there exists power lines for +12 V, +5V, and +3.3V.

The lithium ion battery pack is a component of the power module that is expected to be replaced or tinkered with relatively often. Because of this, there is an increased risk of user error when re-installing the lithium ion battery pack into the power module. To protect the output voltage regulators from an unexpected negative polarity that could be experienced when installing the lithium ion battery pack to the power system incorrectly, a reverse polarity protection mechanism will be implemented. The most common method for doing this is to use a semiconductor component such as a diode or p-channel MOSFET. The MOSFET allows for a very small amount of power to be wasted in the protection circuitry, which is the best for the expectation of a large amount of current to supply the plethora of sensors and motors.

H. Bluetooth

After careful consideration, and the time constraints placed on the group during the COVID-19 pandemic, it was decided to only implement Bluetooth without Wi-Fi into the system. Along with current issues, there were also difficulties with powering the dual Wi-Fi/Bluetooth chips as they required voltages that that were not easily obtained with the current setup of the project. Because of these reasons it was the team's executive decision to omit Wi-Fi and rely solely on Bluetooth for communication purposes.

For this we chose the Texas Instruments CC2640 Bluetooth Low Energy wireless MCU. The CC2540 is a Bluetooth only chip with Bluetooth Low Energy 4.0 capabilities. The CC2540 uses an RF transceiver with its own 8051 MCU. It also has in-system programmable flash memory, 8-KB RAM, and many other supporting features and peripherals. Low-power sleep modes are available as well as short transition times between operating modes. This allows this product to consume very low amounts of power suitable for our project.

The CC2540 requires minimal external components, and comes with reference designs from if manufacturer, Texas Instruments (TI), which allows us to have the documentation required to implement correctly. The chip also has five different modes which can be implemented to keep energy consumption low. These include: Active Mode RX and TX and Power Modes 1, 2, and 3. With a wide voltage range of 2-3.6V this allows for full RAM and register access in all power modes activated. The microcontroller available on the chip has 8KB SRAM and 128-256KB of in-system programmable flash. The peripherals include a 12-Bit ADC with eight channel, one 16-Bit and two 8-Bit general-purpose timers, twenty-one general-purpose I/O pins, USB interface, and a battery monitor and temperature sensor.

Software wise, this part comes with a Bluetooth v4.0 compliant protocol stack for single-mode BLE. This stack comes with the controller and host including: GAP used as the central peripheral, observer, and broadcaster, ATT/GATT used as the client and server, and SMP for encryption and decryption. TI also provides software documentation including sample applications for GAP central peripheral roles, and multiple configuration options.

I. Weather API

In order to provide customers with a reliable and timely execution of the system's main purpose of responding to weather and possible weather threats, the software on the microcontroller will run two main programs. The two programs will run algorithms for parsing the data collected by the sensors and interpreting the Weather Application Programming Interface.

An Application Programming Interface (API), facilitates communication between a server and a database. In the scope of this project, a Weather API will be used to track data of the surrounding forecasts and weather alerts. The Weather API will allow for the optimum opening and closing of the roof in response to weather predictions. The API would solely be used for predictive measures. Using alerts such as lightning strikes and tornado warnings, will add extra security to the customer's structure. A possible stretch goal for the phone application would be to incorporate these warnings and the API to send push notifications to the user on their device about weather and the roof operating statuses.

The basic data the project would need from the Weather API would be: precipitation probability, air quality, lightning strikes, and weather warnings. This data will be pulled from both the current conditions along with a twenty-four hour prediction. The purpose of the Weather API is to make predictions to have the roof react before a storm or hazardous weather conditions reach the structure, and with the weather being very temperament, there is no need to have predictions based twenty-four hours.

To meet all the needs of the weather API, and to provide a timely and effective reaction of the roof, the Dark Sky API and the MeteoGroup lightning API will both be used to predict the opening or closing of the roof to avoid extraneous weather that would not be picked up by the sensors until the storm reached the structure. The limited free subscription allows for adequate request calls to the database, and provides more weather data and more accurate forecasting to the minute.

IV. Hardware Details

The following sections will include a more in-depth breakdown of all hardware components of the system design, as well as an explanation of how each system will be integrated to create the final design. For a better view of how the system operates see Fig 2 below. Due to COVID-19, the pressure sensor was removed from the project.

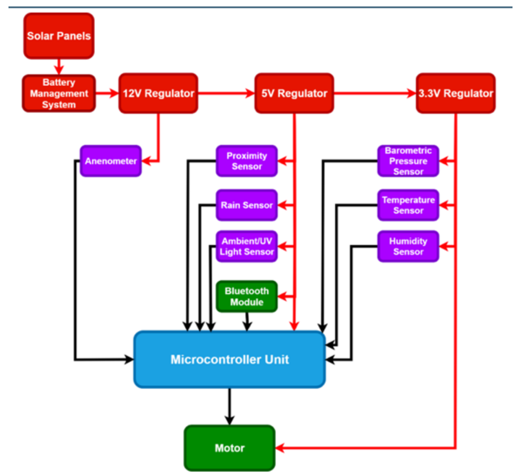


Fig. 2. System overview flowchart

As seen, each sensor requires a different voltage input provided by the power module. After all data is read in from the sensors by the MCU, data is sent to the motor module to control the roof's operation and sent via Bluetooth to update the application for user readability.

A. Microcontroller Unit

After careful consideration and analysis of all hardware and software specifications, the chosen microcontroller for this project is the ATMega 1280. However, because of the COVID-19 crisis, the final design for the microcontroller was unable to be realized, so it was replaced with the DF Robot Bluno Mega 1280 development board. This board is equipped with an ATMega 1280 along with Texas Instruments CC2540 Bluetooth Low Energy chip capable of Bluetooth 4.0. However the use of the development board limits the number of Inter-Integrated communication (I2C) devices the team can use since the board is only equipped with two I2C ports.

B. Sensor Module

The final sensor module is all placed onto one PCB, which connects from the power module PCBs then to the Bluno Mega 1280. The sensors must be kept separate from the bulk of the other components so they can provide accurate readings from the outside of the building. Shown below is the PCB layout for the sensor module, note the additional circuitry required to produce an accurate output is also placed on the board. From there, the jumpers will be used to connect to the development board. All dimensions are in inches.

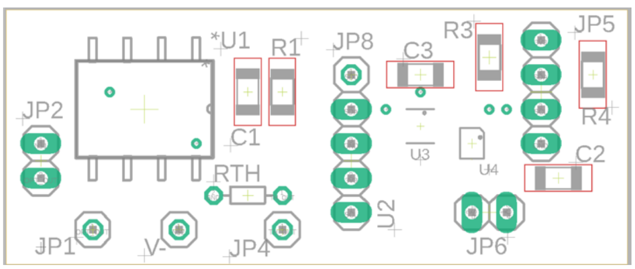


Fig. 3. Sensor PCB layout

C. Power Module

The final layout of the power module will be discussed in the following section apart from the solar panels for possible redundancy. The solar panel used in this system is used as an aid to the battery system for additional renewable energy and will connect to the battery management system before the produced power is sent through the regulation stage.

D. Battery Management

This subsystem includes a lithium ion battery pack, battery charger controller, and reverse polarity protection. Because Lithium-ion batteries have a 3-series configuration of 18650 cells, they have an impressively high energy density that allows efficient storage of power. However, with this, is a charge/discharge cycle that must be managed, which will be done by the MCP73842. The MCP73842 is an integrated circuit (IC) with a voltage range of 8-12V. This linear charge controller provides constant voltage, constant current, cell preconditioning, cell temperature monitoring, advanced safety timers, automatic charge termination, and charge status indication. The designed circuit board was configured following a similar application to that of the datasheet, which is provided in Fig. 7. below.

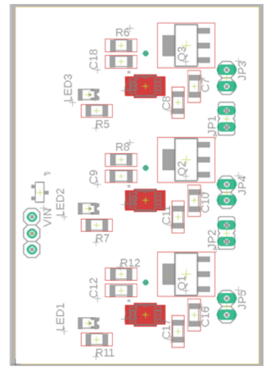


Fig. 4. Battery management PCB design

For the overall A.I.R.E. system, a 12V regulator is required. This regulator needs a 12-20V input for optimal regulation. In order to provide enough power, three MCP73842 systems had to be placed in series to produce the required output. in addition to the BMS circuitry, a reverse polarity protection circuit was also built onto this PCB to protect the system from unwanted damage.

E. Voltage Regulation

The voltage regulation stage of the A.I.R.E. system is provided on a separate PCB shown in Fig. 8. This section includes all remaining details of the regulation requirements including 12V, 5V, and 3.3V regulators.

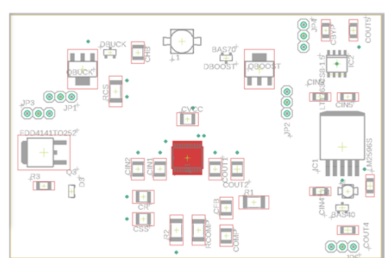


Fig. 5. Voltage regulation PCB

The 12V regulator is the LM25118 buck/boost converter. A buck/boost converter was chosen as there will be a varying input from the lithium-ion battery pack, and it must be able to convert low or high voltage to a steady 12V. With a 3-series lithium ion battery pack, the expected input voltage to the system will vary between 11.1 – 12.6 V. In addition to this, the 12 V regulator, being the main interface between the lithium battery pack and the entire autonomous roof system, it will need to handle the most power flowing through the chip. Lastly, the regulator will need to have an efficiency of 80% or greater with a reasonably low number of external components needed to maintain such efficiency. All of the state requirements led to the choice of the LM25118.

The LM25118 also requires additional transistors to drive the buck-boost controller's output. These transistors are known as high-side buck MOSFET and low-side boost MOSFET. For simplicity, the design in the typical application circuit published in the datasheet will be utilized, as its application similarly matches the A.I.R.E application. This provided design lists the Si7148 N-channel power MOSFET in its typical application circuit published in its datasheet.

The chosen 5V regulator is the LM2596 buck converter. Because it will be receiving the steady 12V output from the buck/boost converter, the regulator will only need to step down the voltage. As its only function is stepping-down voltage, a regulator with high efficiency and minimal external circuitry was preferred. Using the TI WEBENCH® power designer tool, some options were recommended, but the familiarity of the LM2596 and multiple package options offered aided in the final decision.

The 3.3V regulator chosen is the LTI1763 3.3V voltage regulator, which is another step-down regulator similar to the LM2596. Because this component will not be handling a large amount of power dissipated through the package, step-down regulators and linear regulators were considered also using recommendations from the TI WEBENCH® power designer tool. It was found that the linear regulator LTI1763 was 82% efficient compared to 75% efficiency of the step-down regulators. Its small package and minimal external components make for a simple implementation into the system as well.

V. Software Details

The following sections will go into more detail about the programming of the Bluno Mega 1280 and the database structure used, excluding the weather API.

A. Firmware

To program the board the team used the Arduino IDE equipped with a built-in bootloader to burn to the Atmega 1280. The onboard Bluetooth chip is also capable of being programmed via the Arduino IDE as well. By using the Arduino IDE no extra hardware is needed to interface with the board from programming, the board is simply connected to a computer via USB.

The main aspects of the code that run on the microcontroller are the initialization of specific sensors, storing measured weather data, and processing through the logic for and managing the sensor measurements. The humidity sensor and the barometric pressure sensor communicate with the board via I2C. A measurement command must be sent to both sensors to initiate a new measurement and the measurement is read back by the microcontroller via I2C. The thermistor, rain sensor, and anemometer measurements are all based on their voltages which are monitored via ADC.

Every thirty seconds new measurements are recorded from every sensor, except the barometric pressure sensor which is polled every two minutes. The new measurements are stored in a structure and compared to the threshold values that the user sets on the phone application. Some measurements are used in unison to determine different weather conditions such as relative humidity, while other measurements can stand alone to determine weather conditions such as the rain sensor. The specific logic flow that the measurements are processed through is depicted in Figure 10.

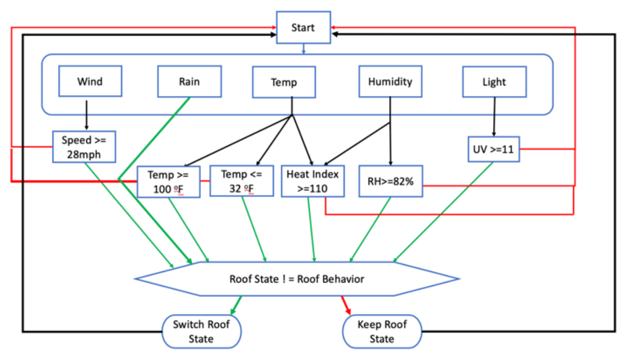


Fig. 6. Software flow chart

B. Database Design

The database will be used to offload larger data from the microcontroller to increase storage space and runtime performance. The two main purposes of the database will be storing user settings and logins, and data for the characteristics and statistics from the API and sensors for the weather. The data from the API will be stored for analysis and statistics for twenty-four hours before being overwritten. This will decrease the amount of storage needed for the database and decrease the time for calculations and data retrieval.

The weather data will be broken into API and sensor tables. The data from the API will be stored for analysis and statistics for twenty-four hours from its original poll request, before being overwritten. This will decrease the amount of storage needed for the database and decrease time for calculations and data retrieval. The data from the API will be used for predictions and analysis, so two tables will be used to break those purposes down. The sensor data will be stored for thirty-six hours before being overwritten, since the sensors are a more accurate measurement of the weather at the actual structure the data is stored for a longer duration.

To reduce the amount of data being collected and the number of pushes to the database only the average, maximum, and minimum value for each measurement will be recorded. The average will be calculated using the current measurement with the stored current average. The values for averages will be reset at every 0:00 of the day. But the data in the database will be stored for thirty-six hours for analysis and statistics that will be pushed to the phone application.

VI. iOS Application Details

The IOS application is one of the biggest features that comes with the A.I.R.E. system, as it provides the user with the following capabilities: Open/close roof, change default attributes, modify accesses and such based on account permissions, and access weather metrics collected by system. It was decided to use Apple's Swift 5.1 programming language for the design, development, and testing phases of the software, as it allowed the team to create a user-friendly interface that provides all the capability desired for this project. It is a powerful and intuitive programming language for macOS, iOS, watchOS, tvOS and beyond, which is exactly the reason it is chosen as the programming language used for this project.

When it comes to the application, the very first page that the user sees is the login page; the user will be asked to login using a unique User Id and password. This will be initially provided by OpenAire to a certain user, who will have authoritative rights from the start and will then be able to create and provide login credentials for application access. These login credentials are a requirement, as application will be inaccessible unless proper authentication is executed. Upon authentication, the user is redirected to a tab-based interface with the following tabs: Settings, Controls, and Weather Metrics.

The application's tabs are meant to do exactly as their name describes. The settings page is made up of two buttons, one that leads to user authorizations page and the other leading to the roof settings page. This roof setting page provides the ability to change the default sensor metric values for which the roof opens and closes. These thresholds are the maximum and minimum temperatures, pressure, wind speed, UV radiation, and whether the roof opens or closes upon rain. These changes are passed via the BLE Serial characteristic provided by the microcontroller. The User authorizations page allows for the current user, if he or she has the proper authorizations, to change the user's name and authorization level stored in the company's database. These changes are then directly reflected in the Google Firebase infrastructure used for this system.

The next two tabs are the controls page and the weather metrics page. In the controls page, a button takes over the majority of the screen. The press of this button effectively leads to the opening and closing of the retractable roof installed. This is a very simple and straightforward page that can be easily updated to control other things based on the customer's needs; this can range from light control, to room sounds control, and more. The last tab is the weather metrics tab. This page shows both the current weather at the location in which the enclosure is found, as well as the sensor metrics being collected at the current moment. The weather data is pulled from Dark Sky API and uploaded to the Firebase database. The IOS device then pulls the data from the cloud and displays it in table view format in the application. The sensor data is pulled from the microcontroller using its BLE serial characteristic.

VII. Mechanical Details

The mechanical design of this structure is based on the design of our sponsor's current projects. The structure design is a 32"x10"x11" structure made from medium density fiberboard (MDF) plywood. The overall goal for this enclosure was make it as small as possible and as lightweight as possible, or under 30lbs. The size constraint was altered slightly as the size of the actuators played a large role in deciding the enclosure dimensions. Shown in the below figure, the top panels include indents for the actuators to fit in, and the plexiglass panels for the roof will be attached to the actuators.

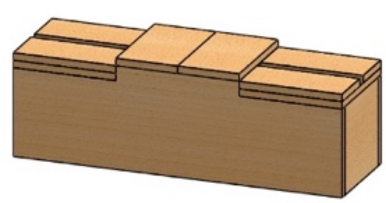


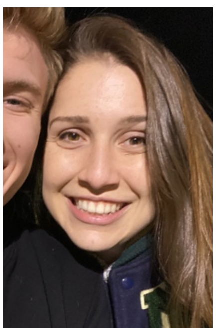
Fig. 7: Solidworks design of enclosure

The solar panel will be attached, at an angle, to the side of the building, and the sensors will attach to the corner of the roof out of the way of the plexiglass panels. All other components that are not required to be outside, will be inside the enclosure away from the roof's opening. This is to protect the essential working components from damage due to outdoor weather conditions.

VIII. Conclusion

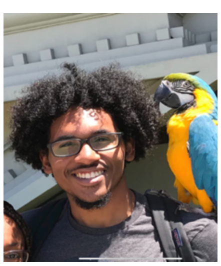
The A.I.R.E. - Autonomous Intelligent Roof Enclosure is an update of the project sponsor, OpenAire's, current design with multiple hardware and software updates and modifications. These modifications include the implementation of sensors, renewable energy, power flow regulation, Bluetooth implementation, and the addition of an interactive application with data displays and roof controls.

Engineers

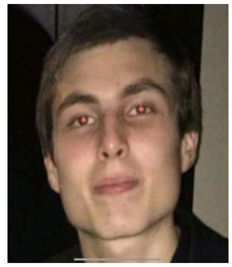


Lauren Miller moved from California to attend the University of Central Florida in 2016. She will be graduating with her Bachelor of Science in Computer Engineering in May 2020. Post-graduation, Lauren plans on moving into a full-time position at Lockheed Martin in an embedded software team, after being in the College Work Experience Program for two years.

Sarah Riseden will be graduating from the University of Central Florida in May 2020 with a Bachelor of Science in Electrical Engineering and a minor in Mathematics. Following this, Sarah will continue to work for Orlando Utilities Company as a full time Electrical Engineer in the Lighting department.



Christopher Smith will be graduating from the University of Central Florida in May 2020 with a Bachelor of Science in Electrical Engineering. Following this, Chris will move on to work as a DLP Applications Engineer with Texas Instruments, Inc.



Ronald Acevedo will be graduating from the University of Central Florida in May 2020 with a Bachelor of Science in Electrical Engineering and a minor in Computer Science. His post college career will take place at Intel Corporation, under the Non-Volatile Solutions group as an SSD Failure Analysis R&D Engineer.

Acknowledgments

The members of this group would like to thank our sponsor, Open Aire, for inspiring and funding this project. Special thanks to Deborah Baker and Donie Riseden for coordinating with us throughout the duration of this project.

References

[a] ZF Electronics, "Magnetic Proximity Sensors (Hall Effect)," MP1021 datahseet, June 2017.

[b] Jacksking, "Jacksking Anemometer, 360 12V-24V DC Pulse Signal Output Aluminum Alloyed Wind Speed Direction Sensor Measurement Tool," B07SN1V427, Oct. 2019.

[c] Sensirion "Datasheet SHT-85", SHT-85 datasheet, 2017 [Revised July 2017]

[d] Little Fuse, "Leaded Thermistors", PR103J2 datasheet, 2018 [Revised Feb. 26 2018]

[e] Texas Instruments, "OPT101 Monolithic Photodiode and Single-Supply Transimpedance Amplifier", Opt101 datasheet, Jan. 1994 [Revised Jan. 2015]

[f] Vishay, "UVA Light Sensor With I2C Interface", VEML6070 datasheet, 2019 [Revised Sept 12, 2019]

[g] ECO-Worthy, "Polycrystalline Solar Panels", ECO-Worthy datasheet 2019

[h] Great Power, "Lithium Ion Battery", PRT-13189 datasheet, July 7, 1028

[i] Microchip, "Advanced Single or Dual Cell Lithium-Ion/ Lithium-Polymer Charge Management Controllers", MCP73842 datahseet, 2013

[j] Texas Instruments, "LM25118 Wide Voltage Range Buck-Boost Controller", LM25118 datasheet, July 2011 [Revised Mar. 2018]

[lk] Texas Instruments, "LM2596 SIMPLE SWITCHER® Power Converter 150-kHz 3-A Step-Down Voltage Regulator", LM2596 datasheet, Nov. 1999 [Revised May 2016]

[l] Linear Technology, "500mA, Low Noise, LDO Micropower Regulators", LT1763 datasheet, 2019

[m] Atmel, "8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash", ATMEGA 1280 datasheet, 2014

[n] Texas Instruments, "CC2460 SimpleLink Bluetooth Wireless MCU", CC2460 datasheet, 2015 [Revised June 2017]