Engineering Smart Windows Using Arduino

Lindsey Joseph lindsey217@gmail.com

Eun Sun Lee elee@whschool.org

Sarah Masters smasters919@aol.com

Carolina Tejada ctejada24@gmail.com

Governor's School of Engineering and Technology at Rutgers University

Abstract

The Window, a common household feature has great potential to become an automated system implemented in homes across the globe. In this project we designed and constructed a prototype of a marketable Smart Window utilizing the Arduino microcontroller. This open-source platform allowed us to use a variety of sensors to evaluate internal and external conditions and adjust the settings of a room accordingly. Our system is proof of concept that demonstrates a cost efficient and novel method of combining window-mounted sensors and to customize the aesthetics of a room and save energy. We furthermore provide analysis of the cost effectiveness of a Smart Window, proving that the window has the potential to be more than what it currently is.

We believe that by using a Smart Window, a user could potentially save over $100.00 per month on air conditioning, $35.00 per month on heating, and $5.00 per month on lighting. This is a significant amount of money that would reimburse the consumer for the money spent on implementing the window within a 2-3 months depending on how many Smart Window systems are installed. In addition, we discovered that more than $1,000 kilowatts per hour could be saved per month by using a Smart Window. This will not only save money, but is also more energy efficient and environmentally friendly.

In all, a Smart Window has the potential to act as an added convenience to the home, as well as to save time, money, and energy.

1. Introduction

In today's world, almost every device in the home is computerized. However, the window, a common and mundane feature of most houses, has been forgotten. Windows have great potential to become the focus of a practical household system. A window can act as a security system, automatically control a room's environment, and save energy while being cost effective. In the homes of the 21st century, the window, although useful, has not been capitalized on in the market. It has the ability to more than what it currently is - which could make it an integral part of life in the future. In the past, before HVAC systems were implemented, families utilized their windows as a way to standardize temperatures inside of rooms as well as to utilize natural light. Once HVAC systems were created, however, people relied less on their windows. In addition, windows were discovered to let in heat from the environment, forcing the house's HVAC system to use more energy to maintain the preferred temperature. In effect, this leads to higher energy and electrical bills which are both unnecessary and preventable.

However, if the window was to become energy efficient and automated, with each household window responding to its environment, the window would surely become a more commonly used item. By integrating a high technology approach to make the entire window system more energy and cost efficient, a window could become even more useful than they already are.

Today, many homes and items are becoming computer controlled and automated. Garages, doors, televisions, and cars are all being designed to respond to a user's preferences nearly seamlessly. This makes life easier, more time efficient, and individually customizable. In this project we have attempted to integrate windows into the slew of other automated household items on the market today.

In order to create a working prototype of such a device, we utilized a microcontroller called an Arduino. The Arduino can read input values from the environment and output values to light bulbs, fans, and buzzers – to name a few. Using these inputs and outputs we can create a system in which the Arduino responds to its environment and can change the settings of a room accordingly. For example, the inside lights of a room can be turned off and blinds can be raised if there is enough sunlight outside to supply the room with light. This saves money on electricity and conserves energy while creating a consistent home environment.

In addition, the settings of an HVAC system can be adjusted in response to the temperature. The window can also act as a security feature that beeps each time the window is opened. This can help prevent burglary, theft, and can notify parents when young children open a window, putting themselves in danger. These features can ultimately help save energy and be more cost effective for the consumer.

The system we have created is the first to combine light, temperature, and switch inputs to automate light bulbs, fans and buzzers. However, although our idea is the first of its kind, a Korean company, LG Decovil, already produces a window that can be controlled wirelessly using a remote control. Their product operates with a system that can open and close a window with a button. Their product is mainly sold in Asia and is different from our Smart Window because it cannot interact with its surroundings. This makes our product particularly unique because it is the first window to utilize sensor data to make decisions for a room and take action. Out window can react to its surroundings, seamlessly changing settings, all while saving energy, time, and money. Because of the Arduino's ability to interact with the window, we believe that we have discovered a way to employ a forgotten item in the home. The creation of a Smart Window will give the typical window increased potential and will help keep up with the growing technological advancements in America.

2. Background

The heart of our Smart Window is the aforementioned Arduino microcontroller. The main idea behind our research and design is to take advantage of the features of the Arduino microcontroller to create a window that can respond to a user's personal preferences and environmental surroundings. The prototype is something that can help the environment by encouraging users to use natural light rather than electricity in their households. In addition, the prototype is automated so all of the temperature and light changes are made automatically. The user doesn't need to think about changing the thermostat, turning off the lights, etc. It can all be done via the Arduino, making the idea of a Smart Window both efficient and marketable.

2.1 The Window

While our project mainly focuses on advancing windows, the actual window has been a part of the standard home for centuries. There are many different types of windows, although the most classic window in the United States is the double-hung sash window. For our prototype we used a casement window, which was the most popular type of window before the sash window was introduced. A casement window is unique because it is usually opened with a crank, lever, or cam handle placed at the bottom of the frame. [1] (See figure 1)

Figure 1: The casement window we used for our project. Has a crank to turn the window to make for easy opening.

We chose to use this type of window because it would take less motor power to turn a crank than to pull open a sash window. Although we did not have access to a motor that would have actually been able to move the crank, a casement window would still have been the best option for a future prototype.

2.2 The Arduino Board

We utilized the Arduino microcontroller, originally designed in 2005 by Italian mechanical engineers to create our Smart Window. The Arduino is an open sourced prototyping platform that is used to create stand alone interactive objects. The board consists of an eight bit Atmel AVR microcontroller that is used to control all of the functions of the Arduino [2]. The software created for the Arduino functions with different operating systems including Windows, Mac, and LINUX and uses a modified version of C in its programs. This code can then be compiled and transferred directly from a computer to the Arduino I/O board via a USB cable. The board must then be powered by a +5V supply to an independent circuit. The board itself is equipped with thirteen digital I/O pins, five analog I/O pins, two ground pins, and +5V volt pins. These pins are then connected to input sensors to provide output actions.

Physically, the Arduino board has 14 digital output and input pins numbered from zero to thirteen. (See figure 2)

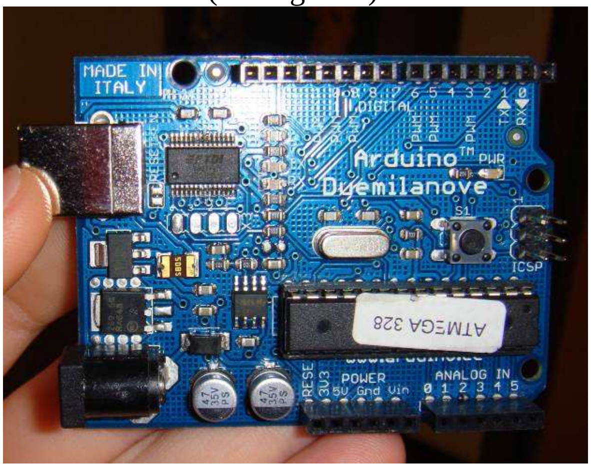

Figure 2: This is a picture of the Arduino Duemilanove board that was used for our prototype. This is one of the most popular types of Arduino boards manufactured.

On the Arduino Duemilanove, pin 13 contains a built in resistor that is necessary when using most electronic components such as voltage divider circuits and LEDs used for our thermistor and photocell. There are also analog output and input pints that are numbered from zero to five. These allowed us to employ a series of sensors that require more than the digital pins have to offer. The analog pins work with a range of values whereas digital works with two functions: on or off. Finally, there are ground and +5V power sources.

All electronics on an Arduino board require connections to the five volt power source and the ground pins in order to operate. Many of these (including the LED) are polarized, meaning that they have a positive and negative lead that must be connected properly in order for the component to function. On an LED, the positive side (anode) is indicated by the longer wire and the negative side (cathode) is indicated by the shorter wire. The positive side is connected to a positive voltage and the cathode is grounded. Sometimes a resistor is needed to reduce the voltage dropped across the LED.

2.2.1 Physical Connections of Board

When building projects with an Arduino, a solderless breadboard or a protoshield can be used to plug in jumper wires, LEDs, servos, potentiometers, accelerometers, switches and other circuit components. On a breadboard or a protoshield, there are numerous rows and columns. The rows of the breadboard are interconnected in rows while the columns are not. These connections are useful when connecting many components to a signal, voltage source or ground.

We attached a number of electrical components to the Arduino pins. Our Smart Window's circuitry includes two thermistors, one ambient light sensor, one photocell, one magnetic switch, a piezo buzzer, a Liquid Crystal Display (LCD) Screen and numerous resistors. The thermistor is used to sense temperature and acts as a resistor. The resistance of the thermistor is measured and each resistance value corresponds to a temperature in degrees Celsius. (See Figure 2)

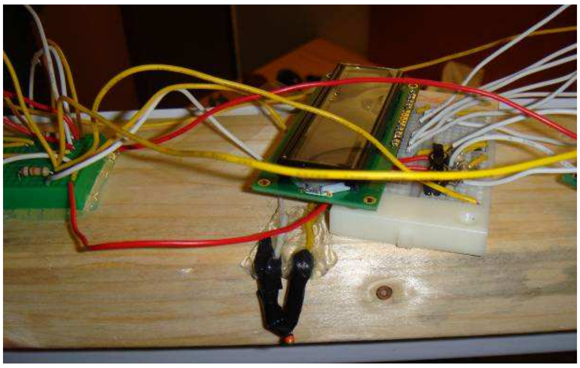

Figure 2: This is an image of the thermistor that was used. It is covered in electrical tape because it has been previously soldered to wire. It is also connected to an LCD screen to display values and a menu.

The photocell and the ambient light sensor are similar to the thermistor because both changes resistance with differing amount of light in the room. The manufacturers define the photocell's dark reading as ~10K Ohms and its light reading as ~1K Ohm. The Ambient light sensor is slightly different from the photocell. The analog pin it is connected to receives a higher voltage with an increase of light. We arranged these sensors so one reads the amount of light outside and one reads the mount of light within the room. Both the temperature and light thresholds can be adjusted by the user.

Finally, for the security feature of our window, we installed both a Roller Lever Switch and a Piezo buzzer on the window. The lever switch is a simple push switch that when pressed down, a circuit is completed allowing for current to flow through. This switch was then connected to the piezo buzzer so that when the switch was open, the buzzer would sound five times and when the switch was closed the buzzer would remain silent. We originally intended to use a magnetic reed switch. The magnetic switch consists of a magnetic cube and a reed switch, that when pulled apart, will open a circuit. The piezo buzzer is incorporated into this circuit so that when

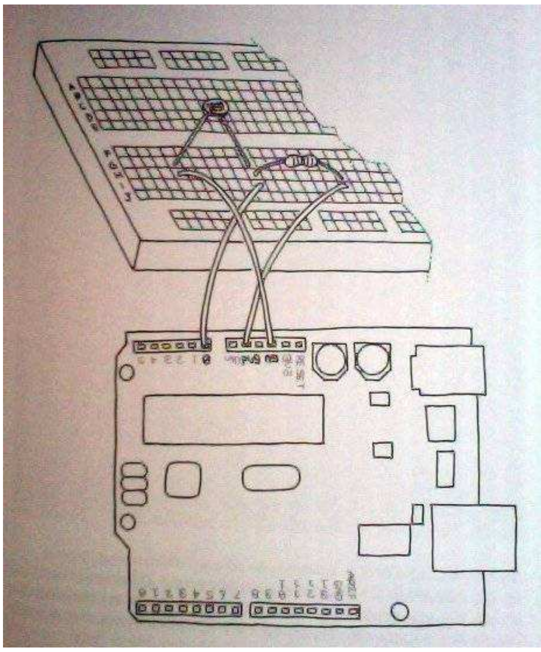

Figure 2: This is a schematic of the photocell wired to an Arduino board using a bread board.

the circuit is open, the buzzer beeps five times, notifying the user that the window is open. The magnet switch is attached to the window so that when the two parts are placed apart from each other – signifying that the window is open – the buzzer will sound. The opposite is true when the switch is connected. We could not use this magnetic switch, however, because at the last minute, the glass on the reed switch cracked, preventing us from being able to use this specific switch. The roller lever switch has the same function therefore using it did not affect how our system worked. Overall, this circuit acts as a miniature security system.

2

2.2.2 Programming the Arduino Board

Each Arduino program is called a sketch and as stated, a form of C is used when creating these sketches. The code's structure includes a series of curly braces {} to enclose the program. Within the braces, there is a setup section and a loop section. The void setup() section, unlike the void loop() function, is performed only once to define the pinMode(s) as either output or input and set variables.

To store values in a sketch, you need to name them as variables in the setup. Variables can be numbers or can represent pins that are attached to LEDs, servo motors, switches or other electronics. This setup function is only read once, and once this part is read, the loop function begins.

Unlike the setup function, the loop function is accessed continuously throughout the program until a delay is put into place. Inside loop functions, values from input sensors can affect how an output object reacts. Commands such as digitalRead, and analogRead are used to read input from various sensors. Commands such as digitalWrite and analogWrite are functions that perform actions such as turning an LED on (high value) or off (low value). If…else statements, and switch statements are placed in the program to read the input values and tell the Arduino what to do with these values. In addition, it is common in Arduino sketches to use delays of a certain number of milliseconds to keep a loop function from repeating too quickly [3].

3. Experimental Design

To create our window design we decided to prioritize the type and number of sensors we would use on our prototype. We realized early on in the project that our full list of possible ideas would not be feasible due to monetary and time constraints. To determine what sensors would be the most useful in creating a Smart Window, we chose sensors that would be the most energy efficient. We believed that making a window that was energy efficient would be the best way to make the window a more efficient tool in the home. In addition, by making our window "Green" we would make it more marketable to those looking to conserve energy in their homes. An energy efficient window would also appeal to those looking to cut costs on their energy bills and save money without wasting too much time. Our Smart Window implements the sensors that we believed would do all of this while still being cost effective.

3.1 Light Sensors

The first sensors we decided to implement were the photocell and ambient light sensor. This was because the light sensors had the most potential to save energy – making it the most marketable part of the prototype. According to our programming, the lights inside of a room turn on and off based on the light inputs from both the inside and outside sensor. The following table describes the various situations light sensors, the light bulb and blinds are programmed to react to:

Table 1.

Inside	Outside	Action of Blinds	State of Light
Light	Light	no change from previous condition	off
Light	Dark	closed	no change from previous condition
Dark	light	open	off
Dark	Dark	closed	on

Table 1: This table shows how the outside light and inside light (light inside of a room) affect how we programmed our blinds and light bulb to respond.

In effect, the ambient light sensor acts as a transistor. The greater the amount of light that hits the sensor, the higher the analog voltage is on the signal pin. These values are then proportioned to light values that are used in daily life. The photocell is slightly different from the ambient light sensor in that it acts more as a resistor. The photocell measures the amount of resistance in different environments. For our particular photocell, the resistance is higher in a dark area and lower in a light area. These high or low resistance values then correspond to useable light values just as in the ambient light sensor.

3.2 Temperature Sensors

The next feature we decided to include was the thermistor temperature sensor because of its potential to conserve energy as well. We believe that its potential to save energy was tantamount to that of the light sensor, so we knew its implementation was key. We also believe that constant temperature readings would be quite useful for the potential consumer because it would provide a way to check the temperature directly on a daily basis.

In order to create an energy efficient window, the temperature sensor would have to be programmed to that it would open a window according to the temperature outside and the user's personal preferences. For example, a user could program the system so that whenever it was hot inside and cool outside, the window would open and the HVAC system would be adjusted. However, we did not have a motor that would be able to simulate this effectively, nor could we connect our sensor to an HVAC system so instead, we connected the readings of the temperature sensor to a miniature fan. This fan was programmed to turn on when the temperature inside of the room rose above 26°C. Otherwise, the fan would stay off as to conserve energy. To do this, we created a code that would turn the fan on when the temperature rose above 26°C in the room. Embedded in this code is a “temperature buffer” of two degrees just in case the temperature inside is fluctuating between two values on the cusp of 26°C. This means that the fan will stay on even if the temperature goes down to 25°C or 24°C. This ensures that the fan will flicker on and off if the temperature indoors is not steady [4]. The point of this system is to bring the conditions of the room to a sustainable temperature. It is to be noted that the system is only used with the inside thermistor while the outside thermistor merely reads the temperature and displays it on the LCD screen for the convenience of the consumer.

3.3 Security Sensors

Another aspect of our prototype is the security system. In some busy cities and towns, crime is rampant and the window is an easy target for theft and forceful entry. Many of these people who live in such communities cannot afford full security systems for their homes and are therefore susceptible to crime. The security sensors that are attached to the window are under two dollars to manufacture, making a security addition inexpensive for almost every potential user. In order for our Smart Window to be marketable to a wide variety of potential users, we believed that implementing a security system would be essential and would broaden the scope of those interested in the prototype. In our program, a buzzer will sound five times whenever a window is opened. With this feature, a homeowner is notified whenever a window is opened. This can also be useful for parents of young children, with a Smart Window, they would be notified by the buzzer when a child unknowingly puts himself in danger by opening a window.

The buzzer is a simple reed switch connected to a piezo buzzer. When exposed to a magnetic field, the two metal components in the reed switch are pulled together and the switch closes. However, when the two metallic pieces are pulled apart, the switch is opened. The piezo buzzer is programmed to sound when the switch is opened.

3.4 LCD Screen

The LCD was the last piece of equipment that we decided to implement in our Smart Window prototype. The LCD serves as an easy way for the user to interact with the features of the window, set their own personal preferences, and read information being sent in through the window (i.e. temperature, light etc). The LCD is also the home of our customizing menu, where the consumer can alter the temperature and light settings they want their window to work with. The LCD works in conjunction with a button system that changes the default settings by pressing the button a different number of times. This piece of equipment makes our prototype user- friendly and marketable.

Figure 5: The LCD screen has a number of contact points (top left). They are connected to input digital pins, 5V and ground. It is a 16x2 LCD module.

4. Implementation Hurdles

The major elements of our window, as described in the Experimental Design, involved a large amount of code to work properly. The following is an explanation of the problems we encountered when writing code for this project and how we solved these problems to create a working prototype.

The first hurdle we encountered involved wiring our prototype to an HVAC system in a room. In order for our prototype to demonstrate the feasibility and usefulness of a Smart Window, we would ideally want out window to control the air conditioning system as well as the lighting in a room However, since this was not possible, we decided to create a built in board that had a light bulb and two outlets with switches attached. From this circuit, we connected our light and temperature sensors to a light bulb and fan respectively. For a marketable window, however, we would have our window connect directly to HVAC and lighting systems already installed in homes.

In addition to the HVAC and lighting problem, we also encountered a few logistical problems when programming code for a menu. The only way to make our model interactive was to ensure that the user could set his or her own personal settings. The menu allows the user to determine what temperature he or she would like the room to be and what amount of light should be present. However, the buttons that we had for the menu were extremely sensitive, so depending on how long and hard the user pushed the button, the menu would scroll down multiple times without stopping at the desired screen. To solve this problem we included a while loop inside of our if-then statement in order to prevent the sensitivity from affecting how the menu functioned.

Another problem we encountered when putting together this circuit board was the power supply. Because the Arduino only supports approximately 5V, most of the appliances we were trying to run (i.e. the fan and light bulb) could not be connected directly to the Arduino without a significant amount of power from a wall socket. For this reason, our group created a box with an independent light bulb and wall sockets and switches. We used one socket for a fan and the switches are manual overrides for the lamp and fan.
The box is plugged into a 120V wall outlet and with relays, operates the light, switches and sockets. They are all connected via wires to the Arduino pins. The first model we made had enough power to turn on the light bulb, but did not have enough power to turn on the fan. In fear of burning out the Arduino board, we built a voltage regulator.

Yet another problem we faced when constructing our Smart Window was the lack of a motor strong enough to open and close the window. The prototype window we used was a casement window that had a crank to open and shut the window pane. Unfortunately, due to budget restrictions we could not purchase a motor that had the ability to open this window. We were, however, able to purchase a relatively strong stepper motor to move the window's blinds up and down in accordance with the amount of light in the room. The stepper motor is programmed to move a certain amount of rotations opening the window because it has the ability to turn both clockwise and counterclockwise. Although we only used the motor for the blinds, the idea of using a small but powerful motor in a future prototype gives the user a small glance into the possibilities and future of a Smart Window.

4. Cost and Energy Analysis

The main reason for creating a Smart Window is to make an average American home more energy efficient. This will eventually save the consumer money on their heating, air conditioning and lighting.

In order to manufacture a Smart Window, a small amount of money is actually needed to assemble the window. One can purchase the parts needed to assemble the Arduino for under $4.00. The thermistors, photocells, and ambient light

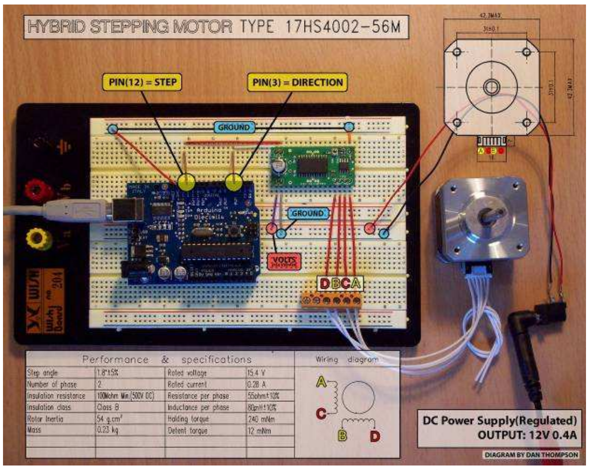

Figure 7: Above is a photo-schematic of the stepper motor's circuit. Our Smart Window utilizes a stepper motor to open and close the blinds.

cells are also just under 2.00 to purchase. The motors to run the blinds are slightly more expensive ranging anywhere from $10.00 to $50.00 depending on the motor power. In order to open the window, the motor would have to be quite strong, making it more expensive. Finally, the wireless communication device, the XBee, is priced anywhere from $20.00 to $30.00. Therefore we estimate that producing a wireless and fully functioning Smart Window would cost anywhere from 70.00 to 100.00. This price is extremely cheap for typical home appliances and is significantly less expensive than a window used in the home. It must be noted that this price does not include the labor, and installation fees that would be added to the overall cost of owning a Smart Window. We believe, however, that these initial costs will be made up in the money that the automated window can save.

In a study done by the United States Department of Energy published by the Wellesley Township, researchers observed how much money it costs to run certain home systems. These systems included analysis on the different costs to run a central and room air conditioning system, a portable heater, and a lighting system.

According to the study done by the Department of Energy, the cost to run a central air condition system for six hours a day for a month requires 900 kilowatts per hour (kWh). [5] Assuming that the cost per kilowatt is approximately $0.1305, the cost to run an air condition system is $117.45 If a Smart Window was to be implemented in a home with a central air condition system, the air flow could be reduced significantly if wired to the sensors on the window. This could potentially save the consumer more than one-hundred dollars. Also, if the user does not have a central air conditioning system but is using a room air conditioner, the savings are also significant. For a room air conditioner that is used for six hours a day for one month requires 270kWh, which translates to $35.34 saved. If the Smart Window was to replace this room air conditioner, the consumer could save a notable amount of money. On the other hand, if a consumer was using a portable heater for eight hours per day, he or she would use 360kWh that would cost approximately $47.00 to sustain. This transfer from artificial to natural heat would save the consumer money yet again.

Finally, the last system that could potentially save money would be the lighting system. Over the years, light bulbs have become more energy efficient and cost effective. However, if a user was to implement a Smart Window into their home and used the natural light for twelve hours instead of using artificial light, they would save approximately $5.00. While this may not seem like a significant amount of money, the numbers can add up tremendously over the years. Therefore if a Smart Window was implemented, it would save the user a great amount of money without having to do any extra work to save this money. In addition, the cost to install the window would quickly be covered by the amount being saved.

5. Related Work

Because a Smart Window is a marketable concept, products are found primarily in industrial rather than academic environments. Windows, while they are not yet being utilized to their full potential, are being still being taken advantage of in the market. Companies are creating more energy efficient and user-friendly windows to sell to customers. Companies such as the Korean manufacturer LG Decovil have created remote controlled windows that can be opened and closed at the touch of a button [6]. This, while not necessarily energy efficient, has the potential to become so if utilized, properly. This technology, however, is mainly useful for the elderly or disabled customers; however, it can also serve as a luxury item.

Similarly, the American company Lutron has also created remote controlled windows that can open and close with a push button switch [7]. Lutron is especially different because in addition to controlling the blinds, Lutron has created windows that are frosted. Lutron uses the idea of "smart glass" that can control the amount of light and heat passing through a window with glazing technology. This is done by using polymer dispersed liquid crystal devices, electrochemical devices, and suspended particle devices.

Electrochemical devices are controlled by a current and they change their opacity based on the amount of voltage flowing through the system. Large bursts of energy are needed to change the window frosting from clear, to translucent, to opaque. This also takes a few minutes for a full process to occur; therefore this technology has only been used on smaller windows (i.e. rearview mirrors, display cases etc). Suspended particle devices (SPD) are also controlled by current but are significantly different from the electrochemical devices. The SPD's are rod-like particles that are suspended in fluid between two pieces of glass or plastic. When a current is applied, the particles align to allow light to pass through. When no voltage is applied, the particles are randomly dispersed throughout the window, making it opaque. The polymer dispersed liquid crystal devices (PLCD), are similar to the SPD's in technology and effect [8].

However, although all of these technologies are viable and add to the effect of a Smart Window, they are all quite expensive, at starting values of $250.00 and upwards. In addition, none of them respond automatically to the outside environment. This is what makes our idea particularly unique.

6. Future Work

Although our prototype incorporated many different automated features, the possibilities of a marketable Smart Window are endless. There were many ideas that our group discussed using, but because of monetary and time constrains, were not able to implement into our window.

Our prototype modeled how light and temperature could be controlled and changed using a single light bulb, blinds and miniature fan that turned off and on based on the input it received from light sensors and thermistors. In reality, if this were to be installed in homes, the window would have to be wirelessly connected to air conditioning and heating systems. In addition, different windows would have to be connected to different inside lights and LCD screens to display information and to manually change settings. Thus, it would be more practical to control the settings wirelessly with multiple Arduinos and XBee wireless modules. [9]

Another useful Smart Window tool would be a wireless remote. We attempted to implement a wireless Wii nunchuck as a remote because of its built in accelerometer, joystick and buttons can be set to control the window's settings. [10] Unfortunately, it didn't interface correctly with the Arduino, and could not be implemented.

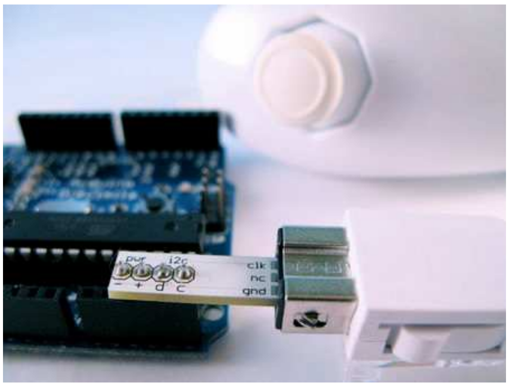

Figure 8: We used a "wiichuck", an adapter for Arduino, when we experimented with the Wii Nunchuck. It is shown here plugged into the Nunchuck.

If a remote were to be created, it would have been able to control the movement of the window and the blinds. Because we did not have access to a significantly powerful motor to open and close the window, our group only focused on opening and closing the blinds instead. A wireless remote would have to potential to do both at the touch of a button.

In addition to having a wireless remote, making the entire circuit wireless was also a possibility we considered when drafting our prototype. XBee, a wireless technology compatible with the Arduino, would allow multiple Arduino boards to communicate with one another. One could be connected to inputs while another is connected to an output, such as the home's HVAC system. This would be especially useful if a Smart Window was to be implemented as a typical household item in the future.

Finally, in order to make the Smart Window as energy efficient as possible, photo voltaic cells could be installed so that the Arduino would run purely off solar energy. The solar cells we had were not rated for the voltage we required. This would make the Arduino completely independent of any power source, and would make the window ultimately wireless. In addition, with some research and development, the solar grids on the windows could potentially store energy that could be used to power other household appliances. This could essentially make the window self-sustainable.

7. Conclusion

The window has the potential to be transformed into a household commodity that is interactive, useful, and energy efficient. The window could be transformed into something that has the ability to harness that natural heat and light for the users' benefit. It's easy to imagine Smart Windows being installed in a home because the technology could cut energy costs for homeowners. If the user chooses to use natural light instead of artificial light when it is available, we estimate that their electricity bills will be cut. On the other hand, if the user desires to keep all sunlight out of his or her house by using the blinds during the day, he or she can program the Smart Window to apply these specific settings. This type of customization can keep Ultra Violet rays from entering the home which can protect expensive furniture from fading. Finally, consumers will be able to use their windows to their advantage by simply allowing the program to monitor the light and temperature rather than operating the light and HVAC systems manually.

In addition to the residential market, a Smart Window could be a part of corporate buildings. In the business world, companies are highly interested in their appearance. Most companies pay extensive sums of money for the upkeep of their building as it is a first impression for the customer. Having Smart Windows that are all programmed to move in unison is both aesthetically pleasing and desirable for companies aiming to impress and modernize their corporate offices. In addition, companies could save huge amounts of money in energy and electrical bills. Thus, companies could actually make and save money by purchasing a large amount of automated Smart Windows.

In all, this working prototype and design project proves that windows have the potential to be more than just an aesthetic part of a building blueprint. When automated, they can improve upon the feature that they were originally designed to provide: light, heat, and ventilation – in a new 21st century way.

9. Acknowledgements

We would first like to thank our mentors, Vincent Lee, our primary mentor, of Columbia University. We would also like to thank and Daniel Hogan of Rensselaer Polytechnic Institute for his instruction and assistance. A special thank you to Blase E. Ur for his dedication to the Governor's School of Engineering and Technology and for his help during the construction of our prototype. Thanks also to Governor's School of Engineering and Technology's Dean Brown and Kristin Frank. We would like to thank the Silver Line and Dominick Benevenga, for donating windows to our project group. Without his help, our prototype would not have been possible. Lastly, thank you to all the sponsors of Governor's School who gave us the opportunity to do this research.

10. References

[1] Casement Window. *Illustrated Architectural Dictionary.* 2008

[2] Shirriff, Ken. 'The Arduino,' in *Arc Language Blog,* June 2009

[3] Evans, Brian W. *Arduino Programming Notebook,* August 2007.

[4] Arduino and Freeduino Knowledge. Ambient Light Sensor, Photocell, Thermistor Code. http://www.freeduino.org/

[5] 'How Much Does it Cost to Run my…' United States Department of Energy, 2008

[6] LG Decovil. http://www.loin62.co.kr/

[7] Lutron. Commercial Products and Smart Glass. http://www.lutron.com/products/

[8] Glass on Web. 'Electronic Smart Glasses' http://www.glassonweb.com/articles/article/192to

[9] ITP Physical Computing. Tymm Twillman's XBee Library for Arduino. http://itp.nyu.edu/physcomp/Tutorials/Tutorials

[10] Arduino with Wii Nunchuck: A Tutorial. Youtube. 2007