Washington University in St. Louis
School of Engineering and Applied Science
Electrical and Systems Engineering Department

ESE 498

Olin Library Seat Status Tracker

By
Sophie Jacobson and Brooke Johnson

Supervisor
Robert Morley

Submitted in Partial Fulfillment of the Requirements for the BSEE Degree, Electrical and Systems Engineering Department, School of Engineering and Applied Science, Washington University in St. Louis

May 2014

Student Statement

This statement affirms that Sophie Jacobson and Brooke Johnson, the designers, have applied ethics to the design process and in the selection of the final proposed design. The designers have complied with the WUSTL Honor Code.

Abstract

The objective of developing the Olin Library Seat Status Tracker was to create a system that would indicate to students the status of seats in the library, occupied or unoccupied. This was achieved by attaching a pressure sensor to the cushion of a chair. The pressure sensor is connected to a Bluetooth enabled microprocessor, which communicates with a main station. Using Processing programing language and interface, a connected seat will either appear as green, indicating that is unoccupied, or red, indicating that is occupied. This system will ultimately be expanded to include each chair in the Olin Library.

Acknowledgements

We would like to thank Professor Morley for helping us design the Olin Library Seat Tracker. We would also like to thank Professor Richter for his ideas and contributions in developing our system. Finally, we would like to thank Professor Trobaugh for letting us use his Android phone charger.

Table of Contents

1. List of Figures and Tables

2. Problem Formulation

2.1 Problem Statement

Every day, students spend time and energy searching for a seat in their university library. Not only do they have to take the time to walk over to the library from their dorm rooms only to find out that all the seats are full, but they must also traverse the floors awkwardly looking for a seat. What if these students could find out the number and location of available seats, without even leaving their dorm room? To solve this need, a system would have to be built into or on the seats of the library, and results would have to be posted online in real-time. To eliminate expenses and make the system compatible with an already constructed library, the system would have to be attachable to pre-existing structures (i.e. a chair, a doorframe, a table, etc). Also, a wireless system would have to be implemented that did not interfere with the Wi-Fi already existing in the library.

2.2 Problem Formulation

With only four months to solve the problem, the system was limited to a single chair. As a realistic goal, it was established that the system would include a sensor, and a type of wireless communication with a pre-existing protocol. Instead of designing a new circuit board, a pre-existing development board would be utilized to solve the problem. From there, unique hardware and source code would be added. By the end of the project, the system would be ready to be implemented to many chairs.

3. Project Specifications

There is a consistent need in the Olin Library for available seats for students. The purpose of this project is to create a system that helps students quickly easily find available seats in the library. The systems needs to be able to identify if a seat is occupied or unoccupied. The target customers for this product would be universities. The customers will need the product to be low cost and low maintenance.

Universities will uses this product to help their students to maximize their study time, so the main users of this product will be students. Students need the interface should be a clear display of occupied or unoccupied seats that can be easily read. They also need a simple way to access this information. They will want the information to be current and accurate. Students will want this technology to work in conjunction with already established facilities, the existing structure of the library, and not interfere with other resources, like the Wi-Fi.

4. Concept Synthesis

4.1 Literature Review

Until now, no other known research has been done with regards to a library seat tracker system. However, the idea was generated based off a similar technique used for parking garages. Large parking garages, such as the garage for Baltimore Washington International Airport use sensors to detect how many spaces are available, and location based on floor and section (Charette). Lights above each space transition from green to red when a space is occupied, which allows drivers to see availabilities from meters away. Additionally, large LED signs have arrows and number counts at the entrance to each floor, demonstrating the way to free spaces. Connecting the sensors to the signs is a central computer that updates availability in real time (Parking Today). The central computer is beneficial because airports are able to monitor occupancy rates and other information associated with the garages.

Inductive loops are the most commonly used detectors for vehicle systems, mostly used at entrances and exits of a garage. When a vehicle passes over an inductive loop installed in the ground of a parking space, currents are induced in the loop which decreases the inductance. A sensor picks up on the change in inductance, and the light above a space changes color. Though inductive loops provide accuracy, they are difficult to install and maintain over time. Research is currently being done on optical sensors that detect passing vehicles (Chinrungrueng). At Blagnac Airport in Toulouse, France, an ultrasonic sensory system is used to detect the occupancy of each space (ITS Decision).

4.2 Concept Generation

After understanding what technologies and types of sensors are being currently used for parking garage smart systems, sensor availability for a library was researched. Sensor options were limited down to optical, proximity, and pressure sensors based on factors such as availability, expenses, and data rate. Advantages and disadvantages for each were evaluated as shown below in Figure 1.

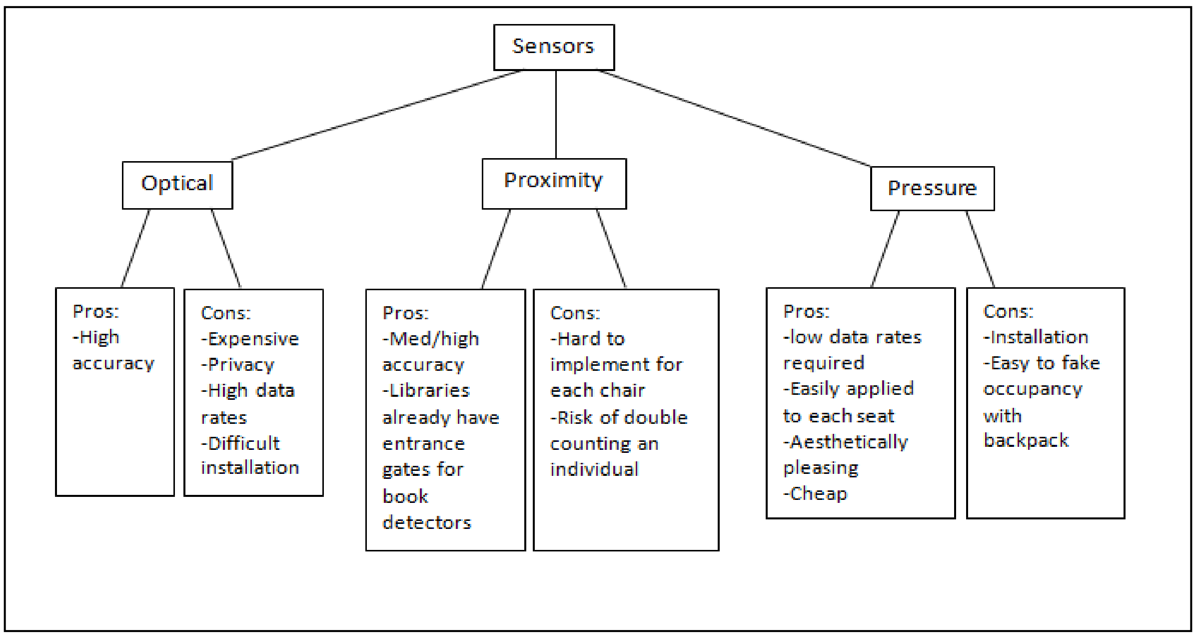

Figure 1. *Tree Diagram of Sensor Alternatives*

For ease of installation and aesthetic purposes, it was decided that making the device wireless was beneficial. Various types of wireless communication were evaluated, and narrowed down to the most ideal types, which include Wi-Fi, Infrared, and Bluetooth. A pros and cons list was also established as shown below in Figure 2.

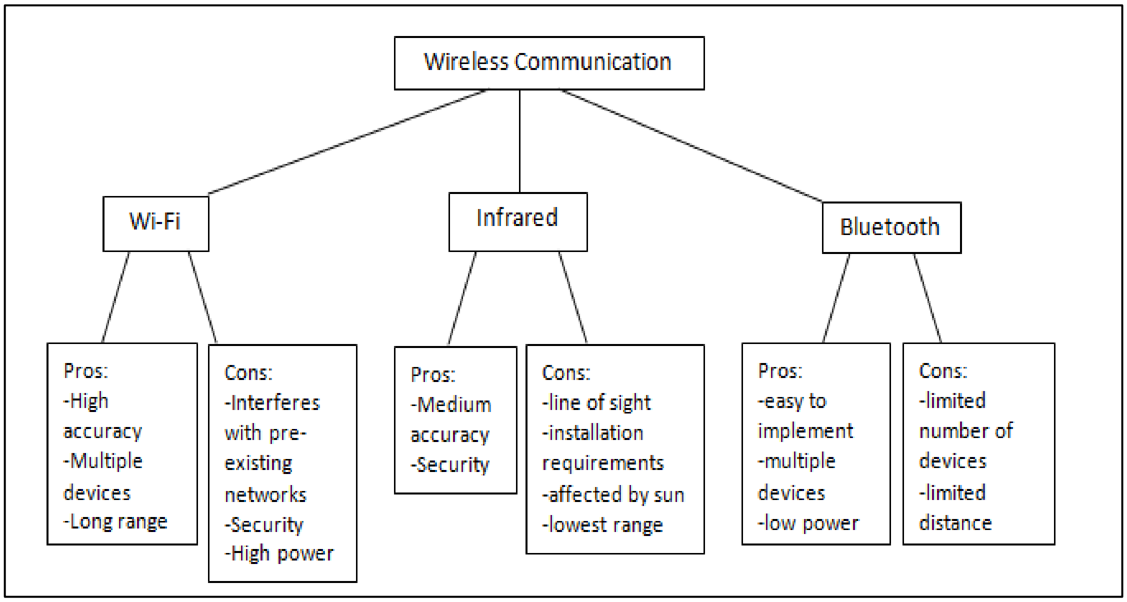

Figure 2. *Tree Diagram of Wireless Alternatives*

4.3 Concept Reduction

After further judgment, with the help of the pros and cons list created above, it was decided that a pressure sensor would be most beneficial. Though optical sensors have the highest accuracy, this means they require high data rates and therefore are expensive. The wiring for optical sensors would also require difficult installation of wires, which is not ideal. This project is intended for installation in a library that is already built, and therefore having to rewire a floor creates a lot of problems. Proximity sensors have high accuracy as well, and could easily be implemented in the entrances of each floor. However, this project is intended for accuracy at the chair level, not simply monitoring how many people are on a floor. If proximity sensors were added to each chair, they could potentially pick up false data, with individuals passing by chairs all the time and not actually sitting down in one.

Pressure sensors, on the other hand, can easily be added to each chair without picking up extraneous signals. Only an individual sitting in a seat can trigger the sensor. Pressure sensors are relatively cheap, and can easily be hidden in a cushion added to the chairs of a library. In terms of installation, wiring of floors would not have to be redone, and only new or reconfigured chairs or cushions would have to be added (which is much cheaper). Also, by making the system wireless, installation is even easier. With pressure sensors, there are low data rates, which make the device cheaper and also allow for easy wireless transmission. The only large disadvantage of the pressure sensor is that a student could fake a chair's occupancy by placing a backpack or book there. However, it was taken into consideration that a student must be saving a seat either for a friend or for themselves while running to the restroom or café. Additionally, the pressure sensor could be calibrated to only trigger at large enough weights. Therefore, with the high ratio of pros to cons for the pressure sensors, it was evident that this type of sensor was optimal for the project.

In terms of wireless devices, it was decided that Bluetooth communication was the best option. Though Wi-Fi has many advantages, there is a risk of interference with pre-existing Wi-Fi networks of the library. Also, Wi-Fi is easy to hack by intruders, which creates a security problem. Infrared communication has higher security, but has the largest list of disadvantages. These include line of sight, interference from sunlight, and a small range. Bluetooth has a larger range of impact, and can have multiple devices linked up to one central computer. This number is limited, but to around 250, so it is not a huge dilemma. A significant factor that impacted the decision was that Bluetooth Low Energy (BLE) chips already exist. Bluetooth was already utilizing lower energy options and low data rates, which was perfect for this project. It was decided that the Bluno, a board that incorporates a Texas Instruments microprocessor used for the Arduino Uno and a BLE 4.0 chip, would be easy to use and code for. With the ease of use and availability, Bluetooth was the communication method of choice.

5. Detailed Engineering Analysis and Design Presentation

5.1 Overview of System Design

A pressure sensor is utilized to indicate whether or not a person is currently sitting in the seat. A Bluno, which is a development board made from an Arduino Uno, with an Atmega 328 microprocessor, and a Bluetooth enabled Texas-Instrument (TI) microprocessor, reads the output from the pressure sensor. The status of pressure sensor attached to the Bluno is reported using

Bluetooth to a central Bluno, the base station. This status displayed on an interface as either a green, available, or red, unavailable, seat. In the ultimate design, only the Bluetooth enabled TI microprocessor would be used, instead of the entire Bluno system, to cut power consumption without unnecessary elements on the board. Additionally, solar cells will be used to power the system.

The setup of the experimental system is described in Figure 3 below.

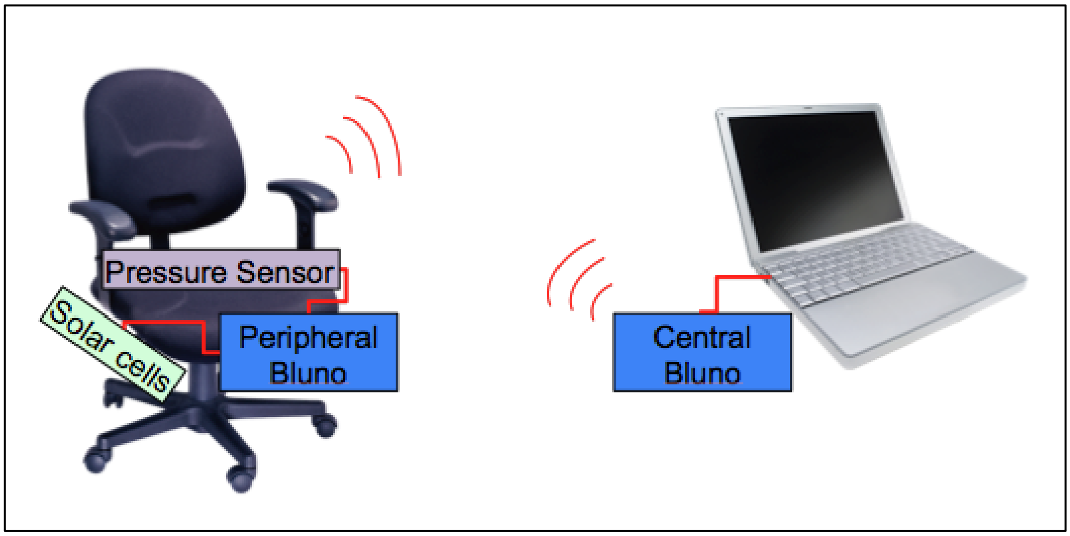

Figure 3. *Experimental Setup of Olin Library Seat Tracker System*

There were several design choices that needed to be made in this system. The system progressed in three phases: the setup of the pressure sensor circuit, the method of powering the system, and creating the interface for the user.

5.2 The Pressure Sensor Circuit

In order to indicate whether or not a person is in the seat, the Bluno reads the signal as high when the pressure sensor is push and low when it is not. The internal pull up resistor on the Bluno was utilized and a voltage divider was created. The Atmega 328 microprocessor has an I/O pull-up resister that ranges between 20k and 50k Ohms. By design, the Bluno system runs at 3 volts. The pressure sensor is a variable resistor that ranges from 1MΩ when not pressed and 100kΩ when fully pressed. The microprocessor reads the input as high at $V_{in} > 2V$, low at $V_{in} < 0.8V$. The pull-up resistor provides an appropriate range to reach these voltages levels for pushed or not pushed. Therefore the pull-up resistor should be turned on and utilized to indicate when the seat is available or occupied. Figure 4 shows the connection of the I/O pull-up resistor, the pressure sensor, and the microprocessor.

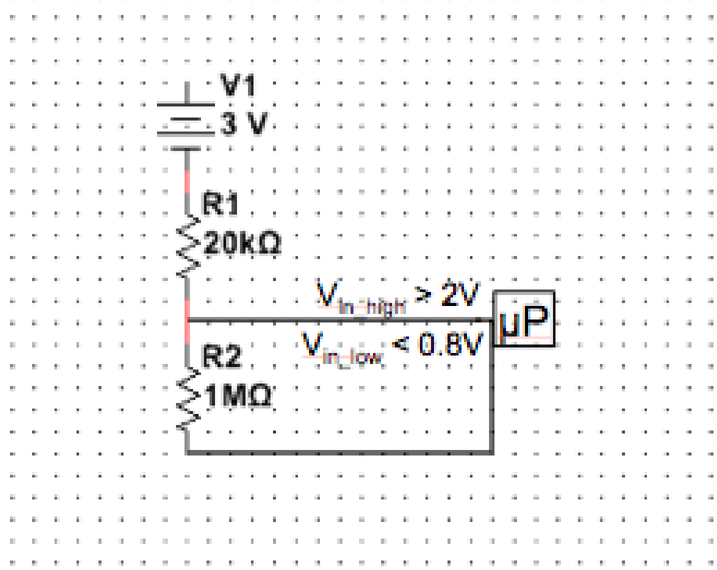

Figure 4. *Voltage Divider Circuit Connected to the Microprocessor*

5.3 Powering the System

In development, a 9V wall wart was used to power the peripheral Bluno and a computer, through a USB connection, was used to the power the central Bluno. Different power-supply options were considered for the peripheral units. Batteries were considered but the lifetime of the batteries in the system was a significant concern. To avoid having to change batteries on each individual unit periodically, solar cells were the alternative that was explored as a power source. Solar cells are both low maintenance for the consumer and sustainable. As a design decision, it was decided that the system would not need to work at late night hours due to the low usage of library seats during those hours. Therefore, the system could remain powered down during those hours when the solar cells are not capturing sunlight.

In the final design, the entire Bluno board will not be used; only the necessary elements from the board will be incorporated into a new surface-mounted chip. The surface-mounted chip will incorporate the TI CC 2540 Bluetooth enabled microprocessor, a capacitor, and connections to the pressure sensor and solar cells. The capacitor will be used to hold charge collected from the solar cells. The microprocessor is designed to run at 3.3 volts but can be run at a voltage as low as 2 volts. The capacitor must only drop 1 volt in order to keep the system operational through transmissions.

When the microprocessor is active and transmitting, it draws 18.6 mA of current. With the other small power draws from the microprocessor when it is in its inactive state, the system is estimated to draw a maximum of 25mA. The amount of information that would need to be transmitted is just one bit: 0 for unoccupied or 1 for occupied. The minimum packet length for low-power Bluetooth transmission is 80 bits, which transmits at a rate of 80µs. To consider worst case scenarios, the transmission time was assumed to be 1/3 of a second, a very large over estimation. This transmission would require a 8µF capacitor in order to have only a 1 volt drop in charge in the system, as dictated by the equation: $C\frac{dV}{dt} = I$. This proves that a 10µF capacitor would be more than enough capacitance for our system.

Finally, the time between transmissions must be considered. The system needs to be sending updates frequently enough that the information displayed is accurate. At maximum power, the solar panel can produce about 100µA (see Figure 5 below). If transmission draws 25mA and takes, at maximum, 1/3 of a second to transmit, the capacitor will be refilled after approximately 250mS, using the equation: $\frac{25mA}{45\mu A} * \frac{1}{3}ms = 252ms$. This proves that it is possible to check the status of the seats every minute and provide an accurate update to the user of the system.

A Sanyo Solar Panel was used in testing and will be the solar panel used in the final design. Figure 5 displays the characterization of this solar panel that will ultimately be used to power the peripheral units. This figure displays the current through the solar cells versus the voltage. The maximum power point is achieved at about 4 volts. Each solar cell is expected to provide 0.5

volts and there are 8 solar cells on this solar panel. This graph was generated in lighting equivalent to that of the library, which is about 250 lux.

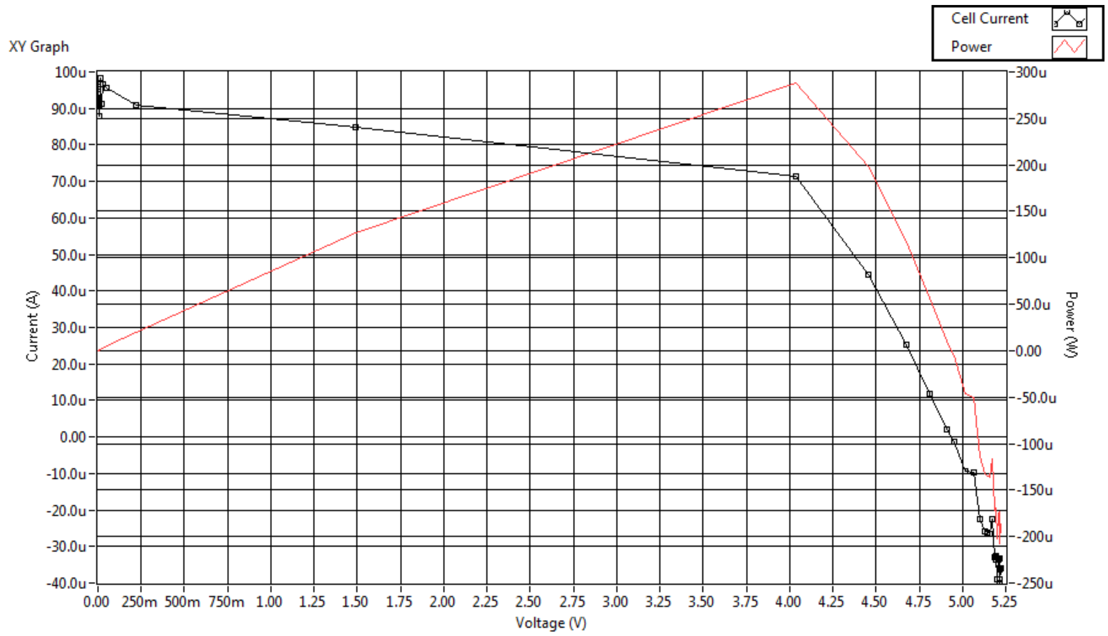

Figure 5. *Current vs. Voltage Curve for the Sanyo Solar Cells*

5.4 The Interface

Processing code was used with Java-style language (see code in Appendix B). Processing is an open source program with many online forums, which also works well with Arduino boards. To first get the two Bluno boards linked up through Bluetooth, a series of steps using Arduino language was performed as instructed on the Bluno Wikipedia page. Once linked with Arduino, programming code was written to model Arduino code. Hence, an Arduino library was imported, and an Arduino object was created. In terms of setup, the Arduino attached to the computer was located with the serial port number, and then the running interface was established to have 4 rectangles all starting with a green color. Also established in the setup was the pull-up resistor. The internal pull-up resistor was called setting the pinMode to input and the digitalWrite to high. Utilizing the pull-up resistor was done in code, instead of using more hardware, in order to make the hardware simpler and cheaper. Also, the input pin is a digital pin, which is important to allow for this pull-up resistor to work. With digital input, there is a binary – the chair is either occupied or unoccupied. In the future, if the weight of a person were to be considered, an analog pin could be used in order to calibrate the pressure sensor to read different weights as different inputs.

In the draw loop, the fourth rectangle that changes colors is created, starting off green and changing pink if the sensor picks up an input (through the use of if/else statements). One may notice that pink is associated with a 0 and green is associated with a 1 which may seem backwards. This is because the pull-up resistor switches high values to 0 and low values to 1. For the sake of this project, only one of the chairs was to change color, but in the future this part of the code could be implemented for all four chairs, using case statements.

After the two boards were synced up through Bluetooth, the processing code was downloaded to the central board connected to the computer, and then sent the code to the other board through Bluetooth to the other. Therefore, when the second board (attached to the pressure sensor) detected a change in sensor output, the computer interface was able to change.

5.5 Final Design

The final design would be designed specifically to fit the Olin Library. The library has five floors, each with approximately 130 seats. Implementing the seat tracker would require 650 peripheral units, at each seat, and 5 base stations, one for each floor. The range of the Bluno system was tested and determined to reach approximately 30 feet. This range is consistent with other Bluetooth enabled devices.

The Bluno boards were ideal for the development of this product; however, they have many extraneous elements that draw power. As described in section 5.3 (Powering the System), the parts needed for the final product of each unit will include: a TI CC 2540 Bluetooth enabled microprocessor, a 10μF capacitor, a solar panel, and a pressure sensor. The central units will consist of TI CC 2540 microprocessors, to receive the signals, and base-station computers.

5.6 Cost Analysis

Ultimately, this product could be marketed to other libraries, therefore the cost analysis will consider ordering 1,000 units of each part. These products were found using digikey.com, sparkfun.com, and ti.com. The prices below in Figure 6, however, reflect prices found using suppliers from alibaba.com.

Peripheral Unit	Cost Per Unit	Cost Per 1,000 Units
Bluetooth Enabled TI microprocessor (TI CC 2540)	$1.50	$1,500.00
Solar Panel	$0.58	$580.00
10uF Capacitor	$0.05	$50.00
Pressure Sensor (Force Sensitive Resistor)	$0.02	$20.00
Total Cost Per Chair	**$2.15**	**$2,150.00**
Central Unit	Cost Per Unit	Cost Per 5 Units
Bluetooth Enabled TI microprocessor (TI CC 2540)	$1.50	$7.50
Total Cost Per Chair	$1.50	$7.50
Total Cost		$2,157.50

Figure 6. *Cost Analysis for 1,000 Peripheral Units and 5 Central Units*

The cost of implementing this system would be $2,157.50, plus the cost of the actual base station computers. This cost is very low and reasonable for a university to purchase to utilize as a resource for the students.

5.7 Bill of Materials

The bill of materials for the final design includes:

1. Texas Instrument CC2540 Bluetooth Enabled microprocessor
2. Force Sensitive Resistor – Square (SEN-09376 RoHS)

    a. Sparkfun Product Number: 9376
3. 10μF capacitor - Panasonic Electronic Components (ECE-A1HKA100)
    a. Digi-Key Product Number: P828-ND
4. Sanyo Energy Solar Panel (AM-1815CA)
    a. Digi-Key Product Number: 869-1004-ND

5.8 Hazards and Failure Analysis

The safety hazards of this project include the physical hardware placed inside seats. The hardware, ideally, will not interrupt normal sitting in library seats. The solar panels will be placed on the side of the chair, which could possibly protrude from the chair and cause small injuries. However, this design is very safe and also environmentally friendly. Environmental considerations were significant in this product because of the potential for many batteries to be used at each peripheral unit. This was considered in this design and solar panels should alleviate environmental concerns.

6. Conclusions

The results of this project so far include a working interface on a base station computer that communicates wirelessly with a peripheral Bluno. In the current test system, the Bluno with the pressure sensor is attached to a wall wart power source and is placed inside a cushion. When a person sits on the cushion, the interface on the base station computer changes from a green square (unoccupied) to a red square (occupied). The communication between the two Blunos is completed using Bluetooth. See the figure below for the results of the interface.

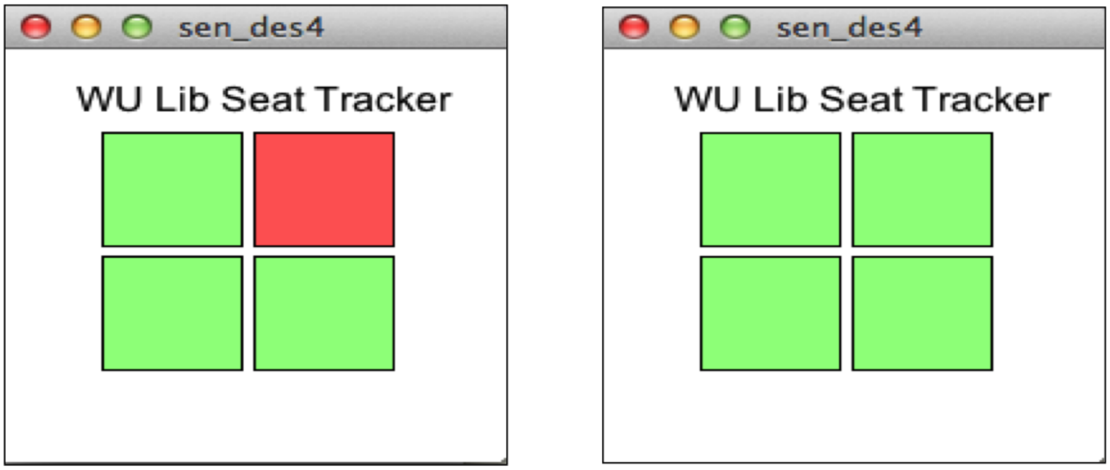

Figure 7. *Interface Results*

The left image shows an occupied seat in the top right corner and the right image shows that same seat as unoccupied. These color changes are trigger by the pressure sensor placed in the cushion of the seat.

The next step of this project would be to develop a website that utilizes the Processing code and updates continuously to display accurate seat statuses. The major complication so far in this portion of the project is implementing Processing code on a webpage. Further research on utilizing Processing in a web-environment will be needed.

To implement this system in the library, a pressure sensor would need to be implemented in each chair. The technology is readily used on one chair and could quickly be applied to many chairs at this point. Eventually, the system would display a map of the entire library floor and display the statuses of the seats. See Appendix B, Figure 3 for an example of this future interface.

Beyond the website, a mobile application could be developed so that students could check for library seats on their phones on the way to the library.

Appendix A: References

Literature Review
http://spectrum.ieee.org/green-tech/advanced-cars/smart-parking-systems-make-it-easier-to-find-a-parking-space
http://www.parkingtoday.com/pluscontent/0205-2.pdf
http://en.wikipedia.org/wiki/Induction_loop
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4090136
http://fresno.ts.odu.edu/newitsd/ITS_Serv_Tech/park_sys_tech/parking_systems_tech_report6.html

Bluno Resources
http://www.dfrobot.com/index.php?route=product/product&product_id=1044#review-title -
http://www.dfrobot.com/image/data/DFR0267/Bluno%20Sch.pdf
http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p_datasheet.pdf

TI CC 2540 Resources
http://www.ti.com/lit/ds/symlink/cc2540.pdf
http://www.eetimes.com/document.asp?doc_id=1278927

Pressure Sensor Resources
https://www.sparkfun.com/products/9376
https://www.sparkfun.com/datasheets/Sensors/Pressure/fsrguide.pdf
http://bildr.org/2012/11/force-sensitive-resistor-arduino/

Solar Panel Resources
http://us.sanyo.com/Dynamic/customPages/docs/solarPower_Amorphous_PV_Product_Brochure%20_EP120B.pdf
http://www.digikey.com/product-detail/en/AM-1815CA/869-1004-ND/2165189

Capacitor Resources
http://www.digikey.com/product-detail/en/ECE-A1HKA100/P828-ND/6935

Appendix B

```
import processing.serial.*;
import cc.arduino.*;
Arduino arduino;
PFont f;
int FSR_Pin = 6; //analog pin 0

void setup(){
  arduino = new Arduino(this, "/dev/tty.usbmodem411", 57600);
  arduino.pinMode(FSR_Pin, Arduino.INPUT);
  arduino.digitalWrite(FSR_Pin, Arduino.HIGH);

  size(200, 200);
  stroke(#000000);
  background(#FFFFFF);
  f = createFont("Arial",16,true);
  textFont(f,16); //size 16
  fill(0); //black font
  text("WU Lib Seat Tracker",30,30); //add text to top
  fill(#99FF99); //creates 3 rectangles
  rect(40,40,55,55); //1
  fill(#99FF99);
  rect(40,100,55,55); //2
  fill(#99FF99);
  rect(100,100,55,55); //3
}

void draw(){
  rect(100,40,55,55); //generate fourth variable rectangle
  stroke(#000000);
  int FSRReading = arduino.digitalRead(FSR_Pin);
 if (FSRReading == 0){
  fill(#FF6666);} //pink
 else if (FSRReading == 1) {
  fill(#99FF99);} //green
 println(FSRReading);
  delay(100); //just here to slow down the output for easier reading
}
```

Figure 1. *Processing code that receives seat status and displays this status on an interface*

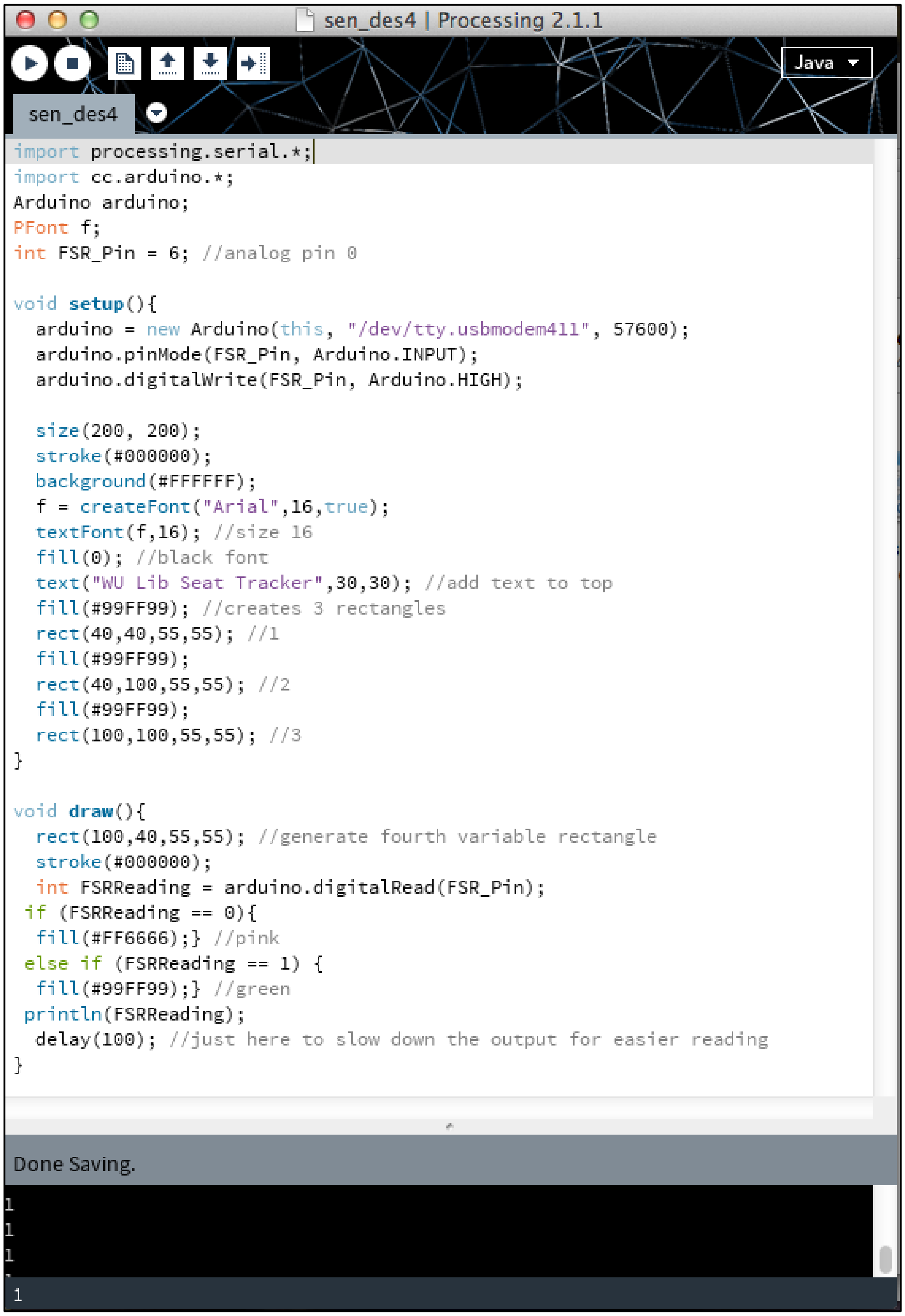

Figure 2. *Processing Code while Running*
This consul displays the status of the seat as a 1, occupied, or a 0, unoccupied (see 1's shown in the bottom of the image).

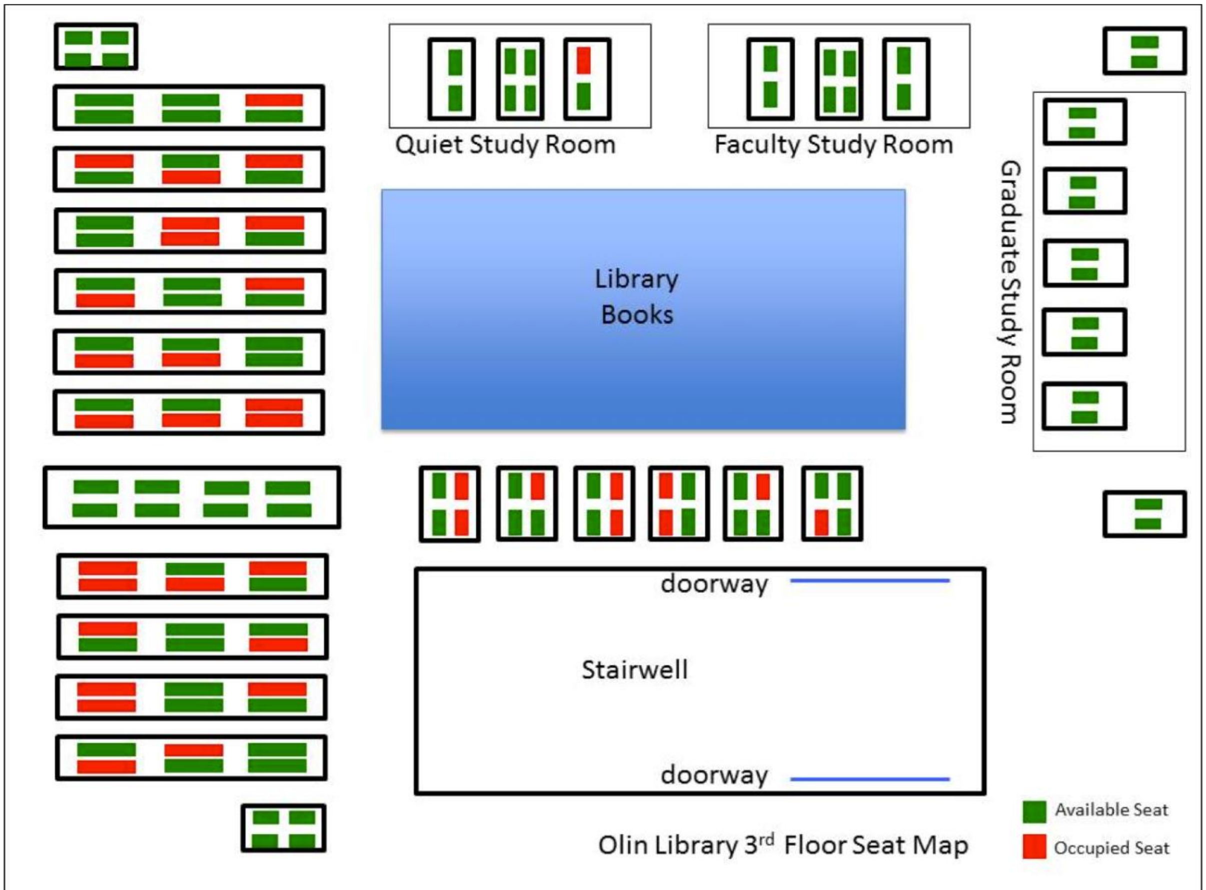

Figure 3. *Future Interface of Website for Olin Library 3rd Floor*