**EEL 4914**

**Senior Design**

**Fall 2014**

**Critical Design Review:**

**Omega Vest**

**Ryan Foxworth**

**Omar Halabi**

**Juan Lopez Marcano**

**Justin Parker**

**Seng Loong Yu**

**Introduction**

 The Omega Vest project is considered to be a success. This judgment is based on the successful completion of all criteria within each of the functional sections of the overall project. These subsystems, as outlined in our test plan, are:

1. Computer application, the Omega Player
2. Microprocessor interface and serial communication
3. Power distribution system
4. Vibration motors
5. Pneumatic compression system
6. Pneumatic scent system
7. Packaging

All requirements necessary for successful operation of the vest were met. By comparison of our initial calculations and simulations to the operation of our discrete and combined circuits we have achieved the critical operating values. The results of all our tests will be included within the remainder of this document.

**Calculations**

We used BJT’s as switching mechanisms because the Arduino does not output enough current for the motors to operate at their rated characteristics.

OFF Mode (Cutoff Region of Transistor, VB = 0 V ‘Arduino Low Pin’)



ON Mode (Saturation Mode of Transistor, VB = 5 V ‘Arduino High Pin’)



For big motors:

,,

For small motors:

, ,

Simulation results:

 

OFF Mode ON Mode

 Another problem that we ran into, which arose with the pneumatic compression system, was that we had to determine the angle of which the servo had to rotate in order to open the air valve. If the servo angle were to small, then the air valve would not be fully opened. If the angle were too large, then stress would be put onto the servo, which could damage the servo over time. By using some basic trigonometry, we were able to figure out how large the servo angle of rotation needed to be.

 To find the angle, we first measured the distance from the air valve, to the servo, which was housed in a 3d printed enclosure. A pulley bar connects the servo to the air valve. Using a digital caliper, we found the distance to be 6.64 millimeters. We then measured the diameter of the servo star, and divided it by two to find the radius, which was 31.54 mm. Then, we use the following equation: Length = Theta(angle) x Radius(in radians). Next, we solved the equation for the angle (theta), which was our only unknown, and the converted the angle in radians to an angle measured in degrees by multiplying the resultant by 180/pi: this yielded angle of 24.1245 degrees. Since the Arduino only uses whole number integers for the angle of rotation, we chose to use an angle of 24 degrees so that no stress would be put on the servo. We did a similar calculation for the pneumatic scent dispenser system. However, it did not work too well due to uncompensated problems that are discussed below in the “Pneumatic Scent System” section.

**Expected Results**

The Omega Player should be a universal computer application that is able to run in new computers as well as in less sophisticated computers. It should read and parse CSV files and be able to play, pause, fast forward, and reverse the video without any noticeable delay. The Arduino’s flash memory should contain the hard-coded instructions and be capable of receiving and handling multiple messages that will control the digital pins with valves of 3.3 V or 0 V. The 6 V AC/DC adaptor that was chosen as the power supply should correctly distribute voltage/current to each component of the Omega Vest. The vibration motors should only vibrate when they receive a signal from the Arduino and should cease to vibrate when they receive a 0 V signal. The air servo in them pneumatic compression system should turn at least 25 degrees to open and close the air valve which regulates airflow. The scent servo should turn at least 35 degrees to press down and release the desired scent to the user.

**Results**

Omega Player

As specified in the Testing Plan Document, the computer application was tested to do the following: 1) It can work with Windows operating systems; 2) It can play all major video formats at adequate frame rates; 3) It can detect multiple Arduinos; 4) It can send six signals to the Arduino in 50 milliseconds or less.

For item 2, multiple videos were downloaded from Microsoft’s sample video website (see <https://archive.org/details/Windows7WildlifeSampleVideo>)" The formats tested were MOV, MPEG, AVI, and WMV. The program opened the videos with no issues. To test the frame rate, we implemented a frame rate counter to count the number of frames per second. We compared the number returned by the function to the values specified by Microsoft’s website. It was found that the measured frame rates matched Microsoft’s reported frame rates.

To test number 3, two Arduinos were connected to the computer used for developing and testing. We added a function to our video player to list every Arduino connected to the computer and give it a unique ID. Using the unique IDs, we tested the functions we created initially, and found that they perform their tasks correctly.

Item number 4 was tested using timers. First, we tested if the signals could be sent and all the instructions performed in 100 milliseconds or less. The program was adapted to log the time it took to perform the said operations and stop if it took longer. After optimizing the program and reducing the number of instructions, the system was tested with a clock of 50 milliseconds. The test was successful.

Item 4 was the only that required the most attention. During the testing phase, we found cross-talk between the Arduino and the USB. The Arduino was used to receive the signals and turn on the components. Also, it was used for logging information that was used to monitor that the data received equaled the data sent. This was the cause of the cross-talk. Therefore, the USB debugging was removed and was switched to debugging via Arduino IDE. After item 4 was completed, item 1 was approached. The completed program was tested in computers that had Windows 7, Windows 8, and Windows Vista that had the appropriate codecs. Originally, we had planned to also test in Windows XP computers, but Microsoft removed support from Windows XP. The program ran as it did in our developing computers, which have Windows 8, and we repeated items 2 through 4 to make sure that the program is operative under the different operating systems.

Vibration Motors



Step Response for Big Motor



Step Response for Series Small Motors

As we can see from the figures above, the big motor settling time is about 2 ms and the small motor settling time is about 0.1 ms. The settling times for the motors are adequate for our system as long as they are much less than 100 ms, which is our tolerance timer in the Omega Player. In the process of the development of the Omega Vest, we found that the small motors that we ordered come with very fragile wires and that one of them was completely broken. Therefore, we modified our design slightly and added two more big motors to compensate for the chance of failure of the operation of the small motors. Additionally, since our small motors have an operating voltage of 2.7 V, we simplified our design by putting two small motors in series powered by the 6 V AC-DC adapter.

Pneumatic Scent System

 When we were in the initial design stage of the pneumatic scent system, we initially planned to take an existing automatic scent dispenser system, and change the circuitry so that it would work for the Omega vest. At that point, we examined the different arrays of scent canisters sold by a few different popular and brands, and chose the brand which could offer us the most versatility, and consumer availability; we wanted our Omega vest users to be able to just go to Wal-Mart, or some other common supplier, to buy the scent refills. Thus, we ended up choosing “Air Wick.”

 As progress was made with the pneumatic scent system, we realized that it would be easier to just design and manufacture our own pneumatic scent system for the Omega vest. Also, we concluded that it would also further signify our engineering excellence by constructing the whole system element ourselves: with the exception of designing and manufacturing the aerosol scent refill canisters. Our new design, comprised of a 3D printed housing for the pneumatic scent system, a servo that was controlled by the Arduino, and an “Air Wick” aerosol scent canister. We chose to use the “Cinnabun” scent canister for our prototype. However, any “Air Wick” scent canister will work with our pneumatic scent dispenser system.

 While the servo angle calculation aforementioned worked well for the pneumatic compression system, we found that we had to do some tweaking when it came to the pneumatic scent system. We found that the head of our servo, which has a star shaped configuration, had to gain momentum (mass x velocity) before it would be able to depress the aerosol scent canister head enough to dispense the scent from the canister. Therefore, we increased the angle so that enough momentum would be generated. To clarify, the servo cannot go from zero to full speed instantaneously. Therefore, we programmed the Arduino so that it would give the servo a “head start,” if you will, before it made contact with the aerosol scent canister head. The pneumatic scent dispenser system ended up working perfectly.

Pneumatic Compression System

During the assembly of the pneumatic compression system, there were limited instructions or resources that helped us understand how it worked. The air valve has three ports: two inputs and one output. The air tank and air pump are connected to the two input ports, and after a few test we discovered that each were attached to the wrong input port. To open and close the air valve, the servo and the air valve had to be in fixed locations. Therefore, we designed housing for both using Autodesk Inventor and 3D printed the part.

Packaging

 When we first designed transistor controller board, it did not look that professional, and it was meant to rest beside the Arduino. However, the second version of the transistor controller board was designed so that it would easily fit on top of the Arduino, while maintaining a sense of visual beauty. The Arduino, and the transistor controller board, will all fit nicely into a professional enclosure.

**Conclusion and Future Work**

We conclude that our project has been completed successfully. All components function as desired and are within expected electrical behavior. Omega Player software works efficiently and is free of bugs. Movie clips can be encoded and watched through the Omega player and responses are felt from the vest. We thank Dr. Raij for his advice throughout the process.

 There is still much more room to advance the capabilities of the Omega Vest. Bluetooth connectivity can be coded in order to eliminate the wires from the motors to the microprocessor. A more powerful and specialized microprocessor can allow for many more features to be included. Other ideas can be a spray of water, an expanded scent system, more motors of multiple sizes, and even a visual component such as varied lighting options. As a group, we are all pleased with what we have accomplished and are confident that we have successfully demonstrated that the goals we have set out for this project have been met.