WIRELESSLY CONTROLLED LIGHT SUIT

Scott Hampshire, Walter Portillo, Neil Oler, Clayton Anderson
Brigham Young University
Provo, Utah USA

Faculty Advisors: Michael Rice, Aaron Hawkins, Doran Wilde
Brigham Young University
Provo, Utah, USA

ABSTRACT

This paper describes the design and implementation of electroluminescent light suits used by the BYU Ballroom Dance Team. The suits are controlled via a radio. The light suit design process comprised suit design and construction, control box design and construction, and a GUI implementation for a transmitting laptop station. These tasks were divided between nine team members and completed over a three-month time period. Each task had its own challenges, but through the hard work and dedication of team members, the project was successful.

INTRODUCTION

The goal of this project was to develop a system that sends commands to an array of suits worn by dancers. A high-level diagram illustrating the concept is shown in Figure 1. Electroluminescent (EL) wire is attached to each suit. The EL wire is controlled via an onboard controller that receives wireless commands from a laptop located somewhere near the performance area. The light sequencing is created on the laptop beforehand by a custom-written program. The program allows a choreographer to easily configure the lights and see in real time what the sequence looks like. Possible solutions to this challenge were narrowed by constraints both visual and technical, as specified by the Ballroom Dance Team. Our team used these solutions as an advantage to save time in the planning stage. In the following sections we will clarify in more detail what hardware and software we used to make this system possible.

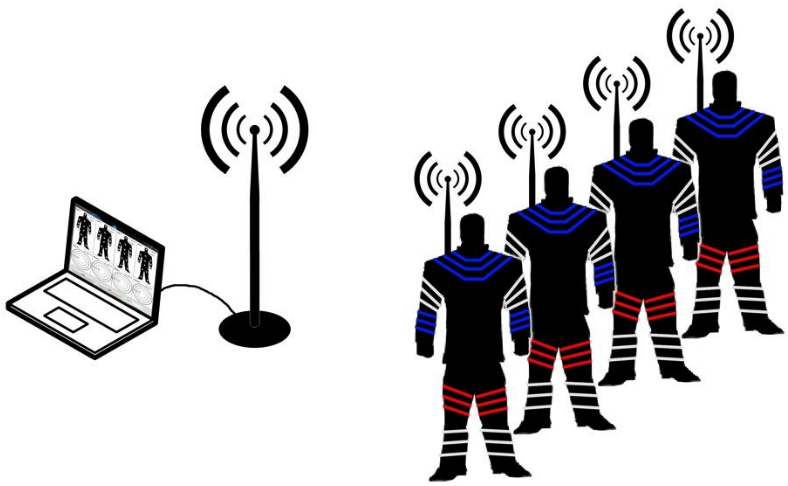

Figure 1: A high-level block diagram of the wirelessly-controlled light suit system.

SUIT CONSTRUCTION

We began the project with a prototype suit that was created as a rough design for the suits to test the functionality/feasibility of the project. This prototype suit was constructed from a bodysuit that had been supplied by the BYU Ballroom Dance Team. On this prototype suit, the electroluminescent (EL) wire was secured to the bodysuit via fishing line. The control box utilized RJ45 connections to interface with the hardware of the suit. Both of which proved difficult to manage on a functional level.

The next-generation suits were a cloth/vinyl hybrid that was custom tailored for each dancer. The shell was made with patches of vinyl affixed to a basic cloth suit. The EL wire was affixed to the vinyl shell of the suits to give rigidity and a firm backing to the wire. This gave a good support on which to mount the EL wire, and allowed the joints to flex without “puckering” the material, which would cause the wire lines to become uneven to viewers. This allowed uniform lines to be traced and routed to give a clean look to the audience. The EL wire was held to the vinyl with an industrial grade rubber cement used widely in the performing arts community.

The suits in their entirety were made of seven different sections: Torso, Legs, Feet, Hands and a Mask. The interconnects between these sections were created using medium molex connectors. The molex connectors were chosen for their durability, versatility and ease of replacement and construction.

Each suit was also paired with a shield that was used during the performance. Shields were made from circular snow sleds that were fitted with a control board, battery, radio and handle for the dancer. We did not face any challenges with the shields on a functional or visual level.

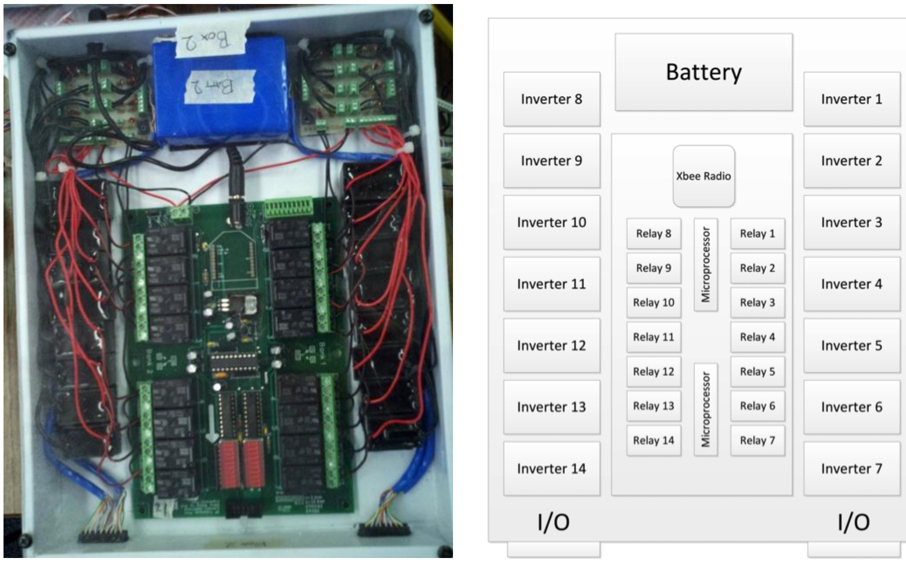

Figure 2: An illustration of the controller box (left) and a block diagram showing the components (right).

CONTROL BOX CONSTRUCTION & DESIGN

Each light suit included a light-control & radio box. These boxes were encased in a cloth pocket on the back of each suit. These boxes consisted of 3 main features: power conversion, wireless control, and hardware encasement as illustrated in Figure 2.

Since the EL wire operates at a 120 volt AC 2kHz frequency, power conversion had to be done if a 12V DC battery was to be used as a power source. The EL wire we used was packaged and sold with a convenient inverter that converts 12V DC to the AC power needed to power the lights. Since the inverters require a load to operate correctly, we had to include a dummy load to keep the inverters from burnout if the lights were disconnected. This commonly happened when a connector failed, or a solder connection between jumper wire and the electroluminescent wire failed. The EL wire has two characteristics to its impedance: it is both capacitive and resistive. So by simply adding a shunt capacitor and resistor corresponding to 1 foot of EL wire, the inverters were prevented from shorting out if the EL wire load was lost.

The control box was also the means for the wireless control. Since the dancers would be moving in the suits, wireless control was the best plan. For reliability, the wireless control was best guarded in the control box mounted on the dancers back. However, radio wave attenuation from the dancers' bodies was a real problem. Experiments with antenna placement showed that a vertical configuration allowed the wireless radio to be maintained at a greater distance from the transmitter without dropping information packets and meet the 500 foot operation distance goal.

The last use of the control box was to house all hardware other than the EL wire itself. In total, the control box housed: the battery, dummy loads, power inverters and control board with the wireless radio control module. The box was constructed out of high-grade polyurethane and was field-drop tested for strength. This was important due to the nature of the use: if the box were dropped or if a dancer fell on the box, it would need to protect the equipment.

COMMUNICATIONS & SOFTWARE DEVELOPMENT

Communication from the controlling computer to the dancers used the XBee radio system. Each suit had a total of two radios to receive the wireless control signals from the transmitter: one in the suit control box and the other in the shield. Each Xbee had two forms of identification: a unique 64-bit serial number and a 16-bit number that is assigned. We used the assigned 16-bit bit number to identify the target because it took less time. The devices and corresponding identification numbers are listed in Table 1. This allowed us to quickly see which suits corresponded to the commands that were being sent via the laptop.

Table 1: Devices and radio identification addresses.

Device	Identifier #
Suit 1	11
Suit 2	21
Suit 3	31
Suit 4	41
Shield 1	12
Shield 2	22
Shield 3	32
Shield 4	42

We used the following as the backbone of the communications:

- XBee Pro Series 1 RF Chip Transmitter unit
- XBee Series 1 RF Chip Receiver unit
- 802.15.4 16-Channels Wireless Relay Controller Board
- USB Modem for XBee Pro Transmitter
- X-CTU Software: This provides the option to read and write XBee parameters, test Baud rate, set identification number and operating mode. These options allow us to configure the application and talk to the microcontroller board.
- Microsoft Visual Studio 2010

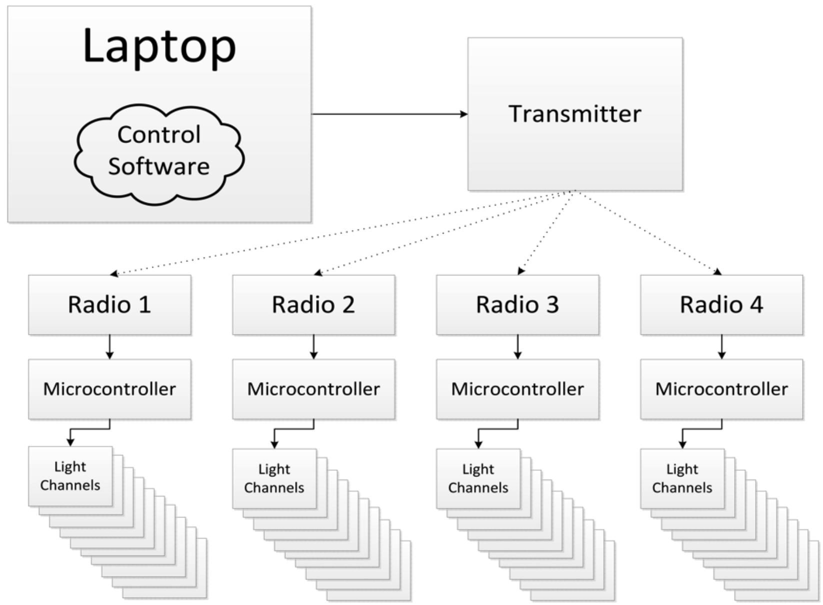

Figure 3: The basic organization of the communication link.

We used Visual Studio 2010/ Visual C# Express because it allowed us to create GUIs easily so that we could develop testing and performance application that could be used for both light sequence testing and debugging with all the suits and shields. It was also convenient because it was easier to send commands in the API mode of the transmitting software through the serial port.

The procedure we followed to develop the communications capability is as follows: We developed in C# the code that was communicating all API commands through the serial port of the host laptop to the transmitter. The code consisted of all the commands that we needed to specify which part of the suit to turn on and for how long. After we correctly configured the XBee chips and the modem to detect the right settings and port, the X-CTU software was opened to test the connection between the computer and the modem chip. There were a few parameters that needed to be checked to ensure that everything was configured properly. Under XB24 was selected in the Modem menu XBee 802.15.4 (10EC version) was selected in the Function Set. Next, “MY parameter” was set in the Network and Security menu folder, “My parameter” was the parameter that allowed us to use the 16-bit channel address. Another important configuration to check was the API. It needed to be set to “Enable” in the serial interfacing folder to “2-API ENABLE W/PPP”. At this point the XBee was programmed with the commands found in the module manual. However, XBee Pro has one parameter that is different from the others: we set it to broadcast mode. We did this by setting the Destination Address Low to ‘xFFFF’.

The API mode had to be used to send commands to specific boards. In this operating mode, API frames are sent which contain the commands to the listening XBee modules such as destination address (MY), which relay to turn on and off and a checksum. API frames start with the byte x7E. The next two byte structures indicate the length and contain the most significant and least significant byte indices i.e ‘0 7’ translate as ‘8’ bytes of information. Next comes the API identifier, which tells the modem XBee which API commands we want it to perform which usually is a 1 followed by a 0 so that no information is sent back to the XBee module. Then, we have the destination address of 2 bytes 0 (board or shield number i.e. 41 is suit 4) followed by a 0. Finally, we send the ProXR command to turn on a relay and the checksum command. The ProXR command to turn on a relay is ‘254 8’ and to turn it off is ‘254 0’. The checksum command is used to check that the preceding bytes were sent correctly. The checksum is the difference between ‘255’ (xFF) the sum of all the numbers after ‘126 0 7’ (this ensures that the checksum always falls between 0 and 255).

API Frame Simplification

Start Command	Length and LSB	API ID	XBee Module Halt Bit	Destination Address	Relay Status	Checksum
126	0 7	1	0	0 41 0	254 8/0	0 - 255

Another important piece of the project was creating a way to send each command so that the desired choreography was displayed. We used a controller that read the choreography information, then interpreted the necessary relay commands, and relayed these commands to multiple relay boards to turn all relays on and off at the right time. Our program read in a text file with the following format:

Minutes	Seconds	Milliseconds	Board Destination	Relay Number	On/Off Relay State
0	1	500	41	4	1/0

This says: after 1.5 seconds turn relay 4 on, and board 41 on. Doing this repeatedly in one text file makes the relay blink on and off every 1.5 seconds. The transmitting program itself actually read in the entire text file and made a series of events that began when the “Start Dancing” button was pressed in the GUI. The program ran through the chain of events until all commands had been sent. For the initial build of the dance sequence, the code was written on a line by line basis which required much “trial and error” to sync the light sequence with the music being played for the dance number.

As stated in the previous section, the program compiles the list of commands when the “start” button was pressed by the user. After the button was pressed and the music for the dance number began, the program began a loop that looked for the indicated time output from Windows Media Player to being executing the code. This time value was included at the beginning of the dance sequence, which reduces the chance of human error in starting the light sequence in the song.

CONCLUSION/LESSIONS LEARNED

It goes without saying that we overcame many challenges during the project. Many of these were the result of inexperience with the technology and a lack of product testing time. The physical suits suffered the most in terms of flawed design and failures. The three most important issues impacting system performance were the following:

- First and foremost was the time needed to lay the wire on each suit. One team member was assigned to the construction of the suits and it was estimated that the required time to trace and lay routes on each suit was 30 hours, this does not include the interconnects between sections of the suit. This could be solved with a simpler wire design or the availability of more manpower.

- Second was the quick deterioration of the wire on the joints of the suit. The constant movement of the dancers resulted in the phosphor layer in the wire breaking down which caused a short to occur between the two conductors. This became very apparent after several practices with the Ballroom Dance Team. Since the project was fulfilled in a short time frame, we were not able to allocate the needed time to test for such failures. These failures were dealt with on a performance basis. It was estimated that 3 hours of repairs were made after each use of the suits in the performances. The EL wire used was not designed to take as much punishment as we intended it to withstand. A revised design that would keep the wire free from the joints would help solve this problem.

- The third major problem that was discovered with the suits carries over from the second problem. When the EL wire shorts, it causes the inverters in the control box to pull a high level of current. This forces the built-in circuit control in the battery to cut power and shut off the control board completely. Although this never occurred during initial performances, it was still considered a major problem that needed to be addressed. After extensive use, the suits began experiencing extensive failures. Although the EL wire can be replaced to provide greater longevity, the long-term solution requires a redesign of the control box and the control circuitry. Changing the wiring of the suit to include parallel wiring in the place of series would not only reduce the visual impact of a failure, it would also lessen the impact of a failed wire.

The flaws in the hardware were the most severe, and could only be resolved with a complete revision of the hardware itself, which for us was not an option. We did, however, face a critical error in the transmission system that was resolved after some critical thinking and a free app from the Android App Store. At several of our performances at smaller venues, it was visually apparent that not all of the instruction packets were being picked up by the radios in the suits. This was obvious because sections of the body would fail to light during portions of the performance. This was nothing that was “mission critical”, but it proved to be an annoyance to us. At a final performance in our largest venue, we discovered that the suits would fail to receive any instructions if they were greater than 5 feet away from the transmitter (instead of the standard 500 ft. range). We realized that we were operating in the congested WIFI spectrum. We used a free app from the Amazon Market to analyze the channel congestion and change our configuration. The analysis is shown in Figure 4.

Figure 4: The congestion in the frequency band used by our radios.

Our radios had been operating on the most congested channel, which was preventing our low power system from picking up the transmission. We changed our system to use channel 14 since the use of this channel for WIFI transmissions is prohibited within the United States. After the changes had been made, all suits were able to perform at maximum efficiency at the required distance.

While the process was tedious and none of the team had experience in the subject, we were able to come up with a design that was both aesthetically pleasing and functional for the Ballroom Dance Team. The lessons that were learned will be applied to future revisions of the suits for the Ballroom Dance Team.