AN LED LIGHT SUIT FOR DANCERS

Andrew McMurdie
Michael Rice (Faculty Advisor)
Brigham Young University, Provo, Utah, USA
Undergraduate Category

ABSTRACT

This paper describes the architecture, design, and performance of a dancing outfit equipped with LEDs and controlled by radio. The lights perform a “choreography” in synchronism with the music. The outfits were worn by a BYU dancing team (the “Cougarettes”) for one of their dance routines.

INTRODUCTION

The Dancers

Multiple dance teams at Brigham Young University, namely the Cougarettes, the Ballroom Dance Team, and the Contemporary Dance Theater, worked in collaboration with the Electrical and Computer Engineering Department of BYU to produce dance outfits that were controlled by radio with variable-color LEDs. It was desired that the lights would change in time with the music, adding a new element to the choreography and presentation of the dance.

The dance team the author worked with was the Cougarettes. Perhaps best known for cheerleading and performing halftime shows during BYU sporting events, the Cougarettes also hold several world championship titles in hip-hop dancing. They continue to perform in national competitions in the U.S., as well as in international competitions across the globe. The challenge given was to develop and produce a suit for the Cougarettes that could be used reliably during their hip-hop routines. Eighteen of these suits were built for the dancers and used in the Spring ‘Cougarettes In Concert’ show over three days at the Covey Center for the Arts in Provo, UT.

Suit Construction Overview

The initial suit design for the Cougarettes (illustrated in Figure 1) consisted of a black top, on which LED strips were to be placed along the shoulders and arms. A custom-built harness was made with 22 AWG wire connecting individual components. There were four wires between all connections: +5V, GND, DATA, and CLK. Connector PCBs with a T shape were used across the shoulders and at the battery joint, where the 90 degree angle bend was
needed. The arms used two LED strips - one on the upper arm and one on the forearm. At the elbow joint short connector PCBs were soldered to the LED strips, and the PCBs were connected by a stretch of wire. This was done primarily to decrease the amount of flexing and bending an LED strip would have to sustain in that area.

![Diagram of the Cougarette light suit](image)

**Figure 1:** Design of the Cougarette light suit top, showing the wiring harness

To control the LED suits, two different custom-designed and custom-built circuit boards were used. The first, called ‘driver boards’, drove the actual LED strips. Each LED strip required one of these boards to be attached. The second board, dubbed the ‘controller boards’, were the brains of the suits, which controlled an XBee radio and stored the timing and control information for the choreography. The controller board sent communications to the driver boards through the DATA and CLK lines of the harness using the I²C bus protocol. When the radio received a ‘start’ command, the controller’s onboard timer would march through the stored choreography information, sending commands to the driver boards as directed.

Powering the suit was a 5V, 2A battery with USB connectors. The +5V and GND lines of the whole harness were tied to this USB connector, thus only a single battery was necessary for the whole suit and all of the LED strips attached.

**Light Choreography Overview**  In order to implement the light portion of the choreography, a standard was built that specifies how the hardware is to store and ‘play’ the choreography in time with the music. This consists of several parts:

- **Pattern** A ‘pattern’ is 32 LEDs worth of color information with one byte per color, for a total of 96 bytes of data. There are 32 LEDs per meter on the LED strips, and it was determined that this would be the longest size of any individual strip used on the suits.

- **Effect** An ‘effect’ is a dynamic change applied to the color. For example, an effect could blink LEDs on and off randomly. The code for the effects is stored in the firmware on the driver board, and each effect is assigned a unique index number. In the choreography information block, this number is what is used, which the firmware uses to call the appropriate
effect code. Following this index, 31 bytes of parameters are available afterwards if the effect has need of them. If not, garbage bytes are written here instead.

**Control Information**  This consists of a two byte address that point to a specific pattern and effect pair. In essence, this is how the choreography ‘action’, or changes, are stored. There are then five bytes of parameters: a special flag byte (for the driver firmware), three bytes of color information (normally used as a default color for the effect), and a single byte timing parameter used by the effect. This information is sent to the driver when the controller sends the ‘play’ command.

**Timing Information**  In order for the light suit to track the music, timing markers are stored in memory in absolute number of milliseconds from the time the song begins. This marker is tied to a set of control information. When the suit receives the ‘start’ command via the XBee radio, the controller’s onboard timer begins to increment. When the counter reaches the timing marker value, the controller will send the linked control information to the driver boards, causing the suit to play the next step of the choreography.

The Pattern and Effect information (a total of 128 bytes per combination) is stored in the driver board’s memory. The Control and Timing information sections are stored in the controller’s memory. All of this information is loaded into the suit before the ‘start’ command is given and playing begins. This minimizes the amount of control the base station has to exert, and reduces the risk of suits reading bad information during a performance that would disrupt the choreography.

**ARCHITECTURE**

**LED Strips**  The LED strips used in suit construction used 16 million color LEDs. Two of the LEDs are controlled by a single LPD8806 microchip, which contains six 8-bit shift registers. RGB color data for the LEDs are stored in these registers as a 7-bit PWM value for each color. When another input bit causes the shift registers to overflow, the output of the last shift register is sent down the LED strip to the next LPD8806 chip. The strips were arranged such that each section of
2 LEDs could be cut and used separately, making it necessary to have an even number of LEDs on any strip for the suit.

The LED strips have four input traces that match the wires in the harness: +5V, GND, DATA, and CLK. (See Figure 2)

To help protect against shorts either from contact, or from sweat, the LED strips are cased in a protective silicone sheath. Both ends of the LED strip were sealed with injected silicone to make this seal as effective as possible.

![Figure 3: A 3D model of the LED Driver board](image)

![Zoomed in side view of LED strip connecting to driver board](image)

**Driver Board** The driver board serves to send power and data signals to the LED strips. Onboard is a PIC18 microcontroller, which runs custom firmware (written in C) enabling various light effects, such as blinks, and stripes that run down the LEDs. It receives commands and data from the controller board via the DATA and CLK lines, which tie into the Molex DuraClick connector (the gray connector on the right side of the board, as shown in Figure 3). Each driver is assigned a unique address on the I²C bus, so that individual light strips can play different patterns and effects.
at the same time. The driver board stores the actual RGB pattern information, as well as a few bytes of control information that the effects code uses to run (for example, the length of the stripe, the color of the stripe, the direction the stripe travels, etc). When the driver receives a ‘play’ command from the controller board via the I²C bus, the effect and pattern specified by the controller are sent to the LED strip.

New effects can be loaded onto the board (as new versions of the firmware) via the PIC programming port (the blue connector in Figure 3). The advantage of building the system this way was that new dynamic effects could be created and tested without needing new, specialized hardware each time. The PIC18 had a limited processor speed, but it was found that even fairly complicated effect code could initialize and run without any noticeable delay relative to the music.

The driver board was connected to the LED strips as shown in Figure 4. 90 degree header pins were soldered onto the four input traces on the LED strip itself.

![Controller Board](image.png)

**Figure 5:** The controller board outside of its housing. Note the ports for the XBee radio.

![Controller Board and XBee Radio](image.png)

**Figure 6:** A picture of the controller board inside its housing, with the XBee radio mounted.

**Controller Board and XBee Radio**  
Like the driver board, the controller board (shown in Figure 5) also has a PIC18 microcontroller, and onboard EEPROM memory. It resides in a custom housing to protect the components, as shown in Figure 6. The controller board is responsible for the major functions of the suit: controlling the XBee radio, storing choreography timing and control information, and sending that information out when playing the choreography.

The XBee radios are Digi International’s XBee 802.15.4 series models operating at 2.4 GHz. Through the XBee radio, the controller receives commands and data information from the base station. The major use of this is in downloading an entire choreography to the suit. During this download, all of the choreography data is fed through the XBee. The controller stores timing and
command information in its own onboard memory, and sends the pattern data through the I²C bus to the drivers as it receives them.

Initially, the firmware on the controller board ran the XBee radio in its transparent mode. Later updates implemented packetization and allowed the use of the XBee API mode. This dramatically improved choreography download time, and resulted in fewer garbage commands accidentally being received and processed by the controller.

SOFTWARE PACKAGES

To program the suits, four software packages were developed. These gave the user the ability to choreograph the lights to the music with a substantial amount of control, without requiring the user to generate large, cumbersome binary files by hand. The programs were developed in Java, and employed the JDOM libraries for XML input/output, and VLC Media Player libraries for MP3 control.

The output of the first three programs was the specially formatted binary file that contained all of the choreography information, which could be downloaded to the controller board of a suit.

Timing Tool The first tool developed was a small utility that allowed a user to load an MP3 song and place timing markers throughout. These markers were simply called ‘beats’. Beats were created by playing the MP3, and while listening clicking an on-screen button. The tool would record the number of frames that had occurred since the beginning of the song, and convert this number to milliseconds.

The beats were shown in a list chronologically, and the user could name the beat to provide helpful information while creating the choreography. Once completed, the list of beats could be saved as a specially formatted XML file.

Choreography GUI The GUI’s main purpose was to allow the user to define the entirety of the light choreography in one simple program. The workflow was broken down as follows:

1. The user would import the XML file produced by the timing tool, which contained all of the song information, the beat list, and beat times.

2. The user would then define the suits to be used by the dancers. This included creating a separate entry for each suit, giving it a name, and listing how many light strips were on the suit. The number of LEDs on each of these light suits was also input.

3. Pattern and effect pairings were created. Patterns could be created by selecting a color-selection widget, which was then used to fill in any of 32 special blocks corresponding to the 32 LEDs any light strip could possibly have. Once these selections had been made, the 31 parameter bytes for the effect could be named and assigned in another specially created window.
4. The final step was to match the beats loaded from the timing tool with the pattern and effect pairings generated in step 3. These were called ‘actions’. The user was allowed to choose if an action was applied to all of the suits or only particular ones. In addition, lightstrips were allowed to change actions independently, allowing for a wider range of effects.

Table 1: All of the timing variables available for defining when a choreography change should occur

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beat</td>
<td>The absolute time of the beat from the beginning of the song.</td>
</tr>
<tr>
<td>Fraction Time</td>
<td>A fraction of the time between actions tied to a suit</td>
</tr>
<tr>
<td>Suit Offset</td>
<td>Specified in milliseconds. Applied to all timing markers</td>
</tr>
<tr>
<td>Suit Scalar</td>
<td>Number between 0-1. Applied to all timing markers for that suit.</td>
</tr>
<tr>
<td>Action Offset</td>
<td>Specified in milliseconds. Applied at the very end for manual timing correction.</td>
</tr>
</tbody>
</table>

To accommodate very precise timings, additional timing constraints could be added (see Table 1). These included a fractional value, which was interpreted to mean an amount of time between suit actions. For example, if a suit had a listed action at beat number five at absolute time 2.0 seconds into the song, and another action at beat number six at 4.0 seconds, using a fraction of 0.4 would mean that the action would not occur until 2.8 seconds into the song. The primary use for this was in so-called ‘group effects’, such as a spiral in which each suit in turn would change to a particular color.

Other timing constraints were an absolute time offset (in milliseconds) that was simply added to the calculated absolute time of the action. This could be either positive or negative. For example, adding a 20 millisecond offset to the example with the fraction above, the action would not occur until 2.82 seconds. This offset was developed primarily to account for some imprecision in the way the timing tool worked – human ears and reaction times could lag behind the actual music beat by tens to hundreds of milliseconds. Also, it was discovered that the activation time of the LEDs required that the action be issued by the controller to the driver boards about 10-20 milliseconds before the change in color/pattern/etc was visible. While this was a small amount of time, for fast songs requiring rapid changes in the lights it was noticeable on a surprising number of occasions. The offset was used to effectively correct this problem.

The final absolute time of the action was calculated by the formula:

\[
\text{actionTime} = ((\text{Beat} + \text{FractionTime} + \text{Suit Offset}) \times \text{Suit Scalar}) + \text{Action Offset} \quad (1)
\]

Once all of these steps had been completed, the information was saved to XML.

**Compiler**  The compiler’s function was to take the XML from the GUI, and convert it into the binary file that the Downloader program would send to the controller board on the suit. In early
versions, it used Equation 1 to generate absolute timing information, and then subtracted to get timing deltas. The timing deltas were written to the binary file because this was what the controller board timing firmware expected. Later updates to the firmware required that the timing markers be written in absolute time, so the deltas were phased out.

**Downloader** The downloader’s function was to take the binary file from the compiler, and send it to the suit via the XBee base station radio. The program used a simple COM port serial connection to communicate with the base station radio, which then broadcasted the data in its transparent mode as simple ascii characters.

In the future, all four packages will be integrated into the Choreography GUI.

**PERFORMANCE**

**LED Strip Reliability Issues** During development of the suits, a large portion of the effort went to ensuring that the LED strips remained intact. It became apparent that the dancers were capable of subjecting the LED strips to stresses that they were never designed to take. Torquing motions would cause the traces on the flexible PCB of the strip to crimp and eventually fail. Any lights further down the chain from these creases would behave erratically, or simply fail altogether. In extreme cases the surface mount components were breaking from the PCB itself, causing the same unpredictable behavior. It was necessary to find some way to reinforce the strips.

To this end, several designs were tested to improve the strips’ durability. The first idea was to use zip-ties to attach a short strip of bakelite to the underside of the light strips along the region where the major IC components were attached. The joints that could be cut (see the edges of Figure 2) were left unstrengthened, as these were the points assumed to have the most flexibility, and the smallest chance of fatal failure. Unfortunately, while this kept more of the IC components in place, the flexible PCB itself was still being torqued to the breaking point, and multiple strips were being broken every practice session.

Worst of all were the areas around the shoulders; when the dancers would lift their arms, the light strip would bubble up from the shoulder surface, causing sharp bends that broke multiple strips. Instead of having the bend applied across the entire section of the PCB near the shoulder, all of the bending was coming from the cutting joints, which failed regularly. Eventually, it was decided to shorten the portion of the light strip attached to the dancer’s upper arm to remove the danger completely.

![Figure 7: The top and bottom of the light strip, with bakelight attached underneath.](image)
The second design iteration was to attach a strip of bakelite underneath the entirety of the light strip. This is shown in Figure 7. While this reduced the flexibility of the suit somewhat, it proved to be the better choice, as strips failed much less frequently.

The third design was an improvement on the second, and addressed a failure issue at the elbow joint. The initial batch of light strips constructed had wires about a half-inch long connecting the upper-arm and forearm strips. This was determined to be too short – the stresses on the joint were breaking the solder connections of the wire straight from the light strip itself, despite the fact that the wires were also held to the board by means of a strong zip-tie. When the wires were lengthened to about an inch and a half, the failures in the elbow stopped almost completely.

**Connector Issues**  
The Molex DuraClick connectors used to attach the bus wires of the harness to the controller and driver boards also exhibited durability issues. The harness wires used crimp pins to attach to the male connector head, and because of the repeated bending and torquing these crimp connections were often failing. The wires would disconnect completely from the crimp pin, which remained stuck in the DuraClick connector head. The solution tested was to apply epoxy to the header once all four of the wires had been attached. This prevented much of the bending at the crimp pin, and resulted in far fewer wires disconnecting in this fashion.

Another problem discovered was that even when properly connected, the male connector could twist a small degree due to the movement of the dancers, and the connection on one or more of the lines would short out for a brief moment. The light strip would then display unpredictable color anomalies that were extremely difficult to reproduce. To solve this problem, a short strip of tape was applied around the male connector, which provided a much tighter seal.

**Radio Issues**  
A communication problem was found early in testing with radio configuration. When the base station sent a start signal, a portion of the eighteen suits would begin playing the choreography, while the others would react as if they had never received the start signal at all. Eventually, the cause was found to stem from two sources. The controller board firmware on the suit was programmed to respond with a basic string when it received a command from the base station. This was received by the base station and displayed for debug purposes. In addition, the XBee target addresses were not set on any of the radios, meaning that the radios would listen to packets coming from any source.

The combination resulted in the condition that when the start command was sent, the suits radios were all responding with the predefined message, and some radios were listening to this response, causing their suits to not hear the start command at all. To solve this issue, the XBee radios on the suits were configured to listen only to the base station radio, and the response signal was cut from the firmware. In addition, the base station was programmed to send the start signal multiple times in succession. The delay from one suit to another, if the suits received and responded to the start signal at different times, was too short to notice.

Another problem resulted from a firmware issue on the controller. There is a UART port on the board that has an input buffer. A programming mistake resulted in the UART being disabled from further input, but the input buffer was not cleared; the board would still process whatever garbage...
happened to be in memory at startup. Occasionally, this garbage happened to coincide with several command opcodes, resulting in a memclear operation. A fix in the firmware resolved the issue.

**In Concert**  The Cougarettes used the eighteen suits in a performance that was hoped would lead to a slot on the ‘America’s Got Talent’ TV program. Due to competition regulations of the show, the dance routine was limited to 90 seconds, and was set to the song “Queen of Wonderland” by the band J*Davey. For an audition, the Cougarettes were required to send in a demo tape of the performance to the America’s Got Talent production team. Sadly, the Cougarettes were not accepted onto the show.

The suits and routine were also displayed at the ‘Cougarettes in Concert’ show (a yearly showcase performance) in late February of 2013. The suits were used in four different shows performed over three days.

While several light strips had to be replaced after the shows, during the performances the suits performed flawlessly.

**CONCLUSIONS**

The suits were a success for the Cougarettes’ performances. Although they were unable to enter the ‘America’s Got Talent’ competition, the routine was well received at the ‘Cougarettes in Concert’ performances. The dance team itself was satisfied with the end product, and post-concert many improvements to the choreography – both dance and light – were discussed, implemented, and practiced. In addition, a pair of pants was actually designed, intended to be used during a second round performance on the TV show. Choreography was never implemented for the top with the pants as a complete whole, but the Cougarettes are excited to be able to use both for shows in the future.

The other dance teams – the Contemporary Dance Theater ensemble and the Ballroom Dance team – built suits of similar design. The same light strips, driver boards, and controller boards were used, although the harness designs were different to accommodate the different costumes. The Contemporary Dance Theater costume was built around a set of pants, with a long strip going down the length of one leg, and rings of lights around the other. This was used for an interesting performance based on a Dr. Seuss book.

The Ballroom Dance Team built skirts with long strips of LEDs going down the side, attached to a central belt for a harness. The costumes were used in the British Open Championships in Blackpool, England at the end of May of this year, where the dancers won first place in the formation category. In addition, the team set a record for being the first formation category team to receive a standing ovation in the competition’s 93 year history.

Many avenues remain for future work on the suits for potential performances. It remains to be seen how future dance routines will be able to employ the dance suits, integrating light choreography with dance choreography in new and fascinating ways.