



A super sparkly, super strong, super dull, Unicorn trap.
-Jeff Eliason

JELLYFISH COSTUME DISPLAY

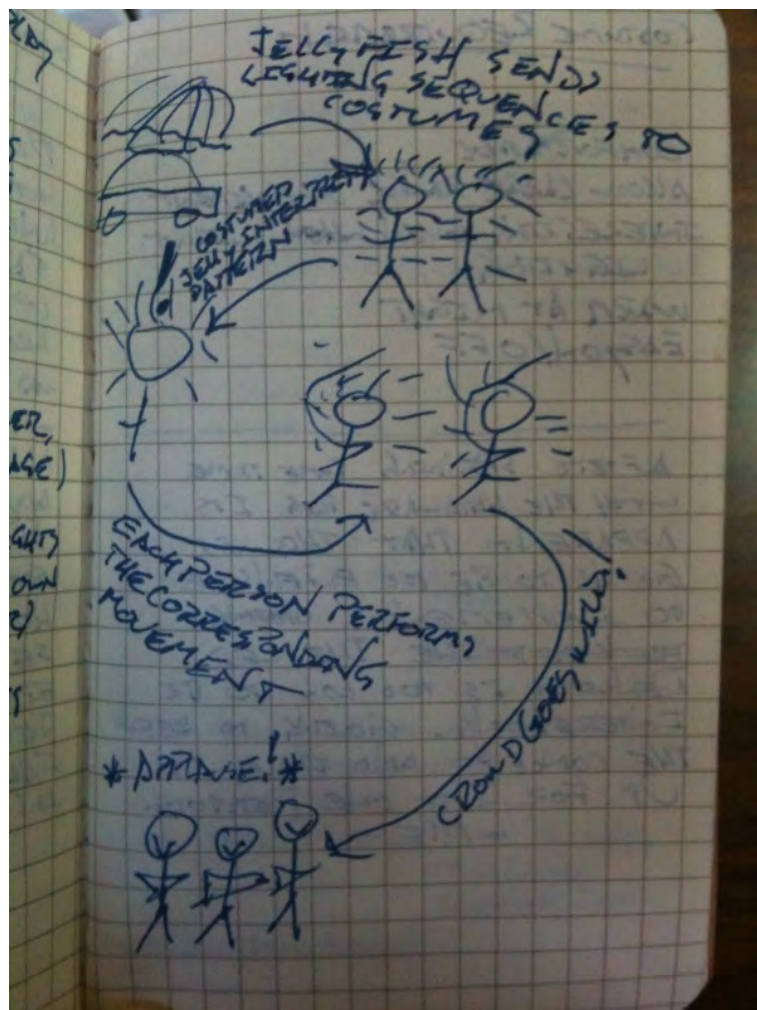
TO INTEGRATE WIRELESSLY CONTROLLED LIGHT DISPLAYS INTO WEARABLE COSTUMES THAT BECOME PART OF THE JELLYFISH DISPLAY MEDIUM

CORE COMPONENTS:

HARDWARE I2C - (MICROCONTROLLER, WIRELESS RADIO, LIGHT PACKAGE)

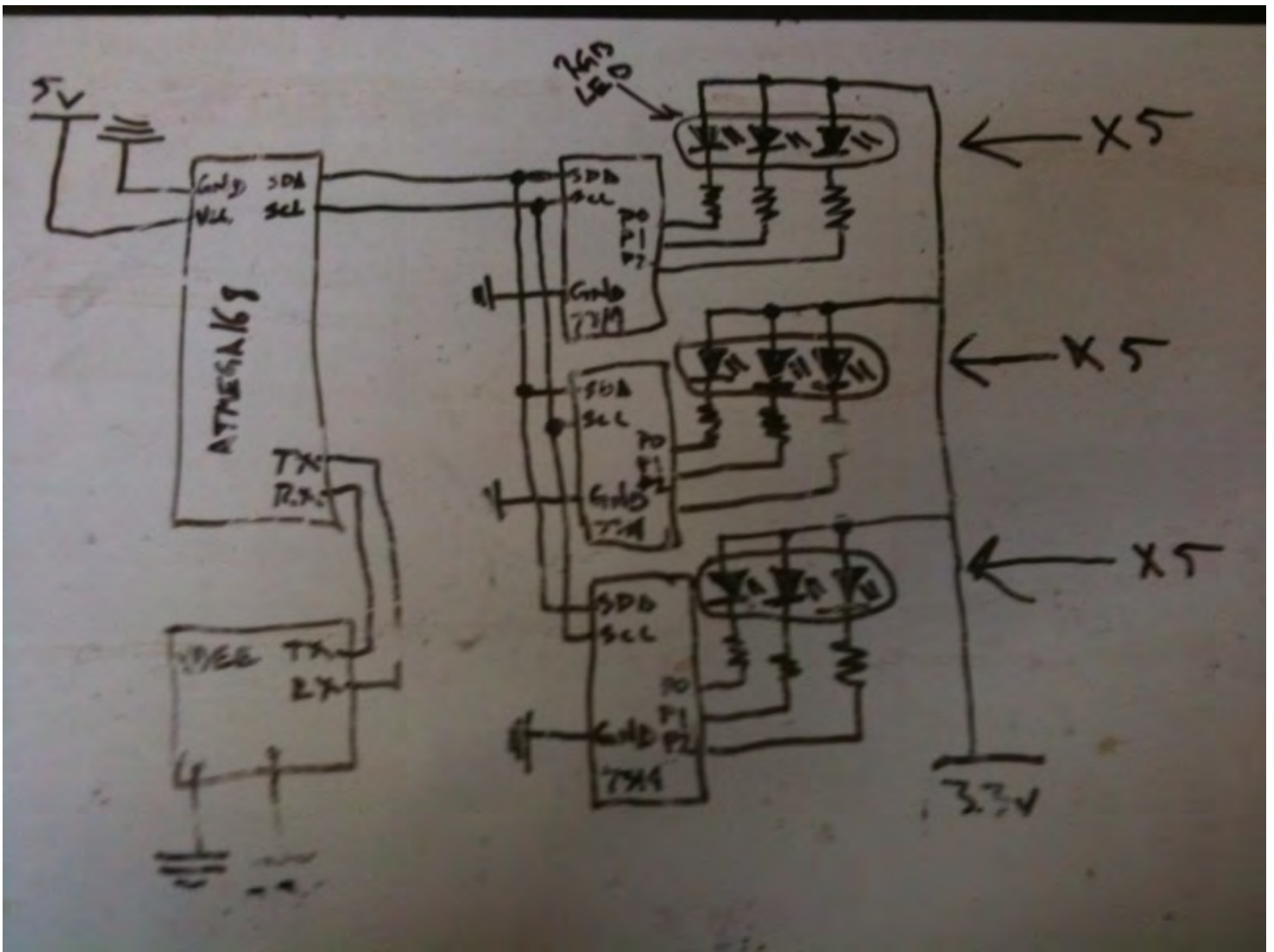
COSTUME - (HOW DO THE LIGHTS BECOME A COSTUME IN THEIR OWN RIGHT, WHAT IS THE THEME, ETC)

MOVEMENT - WHAT MOVEMENTS CORRESPOND TO WHAT LIGHT SEQUENCES, MUSIC?



When I first envisioned this project it was to use the Jellyfish 12000 as a base station. The costumes were to be an extension of the led system on that art car and so the initial requirements were set up around it.

It occurred to me that once you have a mobile led display, it could benefit from choreographed movement that would help maximize the wow factor of costumes displaying their glittering lights in unison or coherent patterns. Just like movement set to music, this would be movement set to music and lights.



After reading (and reading and reading for about 45 days) the datasheet on the MAX7134 LED drivers I drew up the first schematic. Surprisingly it turned out to be correct on the first go.

MAX7314 LED DRIVER

REGISTER COMMAND

0x0F ~~0x00~~ GLOBAL OFF

0x06 0x00 P0-P7 OUTPUT

0x07 0x00 P8-P15 OUTPUT

~~0x0E 0x00-FF MASTER BRIGHTNESS~~

0x10-17 0x00-FF P0-15 PWM

0x02 TO 0100 FOR FULL PWM

0x03 ADDRESS TO MASK FOR OFF

SLAVE ADDRESS BLOCK

SCL 0x60 - YELLOW

SDA 0x64 - BROWN

GND 0x20 ← Not used in this project

V+ 0x24 → RED

WIRE BEHIND TRANSMISSION (SLAVE ADDRESS)

-WRITE SEND (PORT ADDRESS);

-WRITE SEND (DATA);

-WRITE SEND (DATA); - AUTO INCREMENTS TO THE NEXT COMMAND PORT

June 4 2009 1700

(123) MCP+~~100~~ 3302E

3.3 V REGULATOR

250 MA OUTPUT CURRENT

LED DRIVERS (MAX7314)

USE ABOUT 60 MA UNDER LOAD

ARDUINO PINS - I2C

ANALOG 4 - SDA - ~~orange~~ BROWN

ANALOG 5 - SCL - YELLOW

100 OHM - RED 56 OHM - GREEN/BLUE

LED DRIVER PART # MAX7314AEG

MAXIM-IC.COM

DOCS: MAXIM APPLICATION NOTE 4273

MAXIM APPLICATION NOTE 2199

[9-BIT SLAVE ADDRESS] [8-BIT COMMAND] [8-BIT DATA]

Once the hardware was working I had to figure out the software. I had no previous experience with hardware addresses so this was a huge challenge for me. I kept reading the datasheet over and over and finally had an a-ha! moment one day on the way to work.

The shot on the left details the foundation of the hardware control. These are the initial setup values to make the drivers do what I want.

The shot on the right contains the notes on the vreg I use to drop the 5v source down to 3.3, various part numbers, and the colors of wire I chose for each of the I2C pins. At the bottom you can see how the LED driver sends commands to the pins.

To make the LEDs more interesting I attached simple vinyl tubing to the lamps and experimented with stuffing them with fiber optic.

In the pic on the right you can see the completed prototype, including the XBee, micro controller and the LED drivers.

In the picture below you can see it lighting for the first time. I covered the base of the LEDs with electrical tape to manage their brightness. the green LED to the right was a simple test to make sure a troublesome connection was working .

